16.12.2018

Emilia Wąż 260797

Inteligencja obliczeniowa

Zadanie domowe 2 - Zgłębianie danych

Telco Customer Churn

Wstęp

Baza Telco Customer Churn zawiera informacje o rezygnacji klientów z usług firmy Telco. Posiada 21 atrybutów, między innymi płeć, posiadane usługi, koszty miesięczne i całkowite. Klasą jest Churn, czyli rezygnacja, z możliwymi wartościami Yes lub No.

Badanie danych opiera się na predykcji czy klient zrezygnuje z usług oraz co może wpływać na rezygnację.

Informacje o atrybutach oraz oznaczeniach nadanych podczas obróbki danych:

customerID
Customer ID
Random string from concatenated letters and numbers.

gender
Whether the customer is a Male or Female (Male, Female)
gender
Female-1
Male-0

SeniorCitizen
Whether the customer is a senior citizen or not (1, 0)
SeniorCitizen
0
1

Partner
Whether the customer has a partner or not (Yes, No)
Partner
Yes-1
No-0

Dependents
Whether the customer has dependents or not (Yes, No)
Dependents
No
Yes

tenure
Number of months the customer has stayed with the company
tenure
min: 0
max: 72

PhoneService
Whether the customer has a phone service or not (Yes, No)
PhoneService
No-0

Yes-1

MultipleLines
Whether the customer has multiple lines or not (Yes, No, No phone service)
MultipleLines
No phone service-0
No-0
Yes-1

InternetService
Customer's internet service provider (DSL, Fiber optic, No)
InternetService
DSL-1
Fiber optic-2
No-0

OnlineSecurity
Whether the customer has online security or not (Yes, No, No internet service)
OnlineSecurity
No-0
Yes-1
No internet service-0

OnlineBackup
Whether the customer has online backup or not (Yes, No, No internet service)
OnlineBackup
Yes-1
No-0
No internet service-0

DeviceProtection
Whether the customer has device protection or not (Yes, No, No internet service)
DeviceProtection
No-0
Yes-1
No internet service-0

TechSupport
Whether the customer has tech support or not (Yes, No, No internet service)
TechSupport
No-0
Yes-1
No internet service-0

StreamingTV
Whether the customer has streaming TV or not (Yes, No, No internet service)
StreamingTV
No-0

Yes-1
No internet service-0

StreamingMovies
Whether the customer has streaming movies or not (Yes, No, No internet service)
StreamingMovies
No-0
Yes-1
No internet service-0

Contract
The contract term of the customer (Month-to-month, One year, Two year)
Contract
Month-to-month-1
One year-2
Two year-3

PaperlessBilling
Whether the customer has paperless billing or not (Yes, No)
PaperlessBilling
Yes-1
No-0

PaymentMethod
The customer’s payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
PaymentMethod
Electronic check-1
Mailed check-2
Bank transfer (automatic)-3
Credit card (automatic)-4

MonthlyCharges
The amount charged to the customer monthly
MonthlyCharges
min: 18.25
max: 118.75

Podzielone na kategorie 1-4.

TotalCharges
The total amount charged to the customer
TotalCharges
min: 18.8
max: 8684.8

Podzielone na kategorie 1-4.

Churn
Whether the customer churned or not (Yes or No)
Churn
No-0
Yes-1

Obróbka danych

Pierwszym krokiem podczas obróbki danych było sprawdzenie czy wszystkie rekordy pasują do kategorii lub wartości liczbowych atrybutów.

```
# Check rules for all attributes #
E <- editset(c("gender %in% c('Female','Male')",
               "SeniorCitizen %in% c('0','1')",
               "Partner %in% c('Yes','No')",
               "Dependents %in% c('Yes','No')",
               "tenure >= 0",
               "tenure <= 72",
               "PhoneService %in% c('Yes','No')",
               "MultipleLines %in% c('Yes','No','No phone service')",
               "InternetService %in% c('DSL','Fiber optic','No')",
               "OnlineSecurity %in% c('Yes','No','No internet service')",
               "OnlineBackup %in% c('Yes','No','No internet service')",
               "DeviceProtection %in% c('Yes','No','No internet service')",
               "TechSupport %in% c('Yes','No','No internet service')",
               "StreamingTV %in% c('Yes','No','No internet service')",
               "StreamingMovies %in% c('Yes','No','No internet service')",
               "Contract %in% c('Month-to-month','One year','Two year')",
               "PaperlessBilling %in% c('Yes','No')",
               "PaymentMethod %in% c('Electronic check','Mailed check','Bank
transfer (automatic)','Credit card (automatic)')",
               "MonthlyCharges >= 18.25",
               "MonthlyCharges <= 118.75",
               "TotalCharges >= 18.8",
               "MonthlyCharges <= 8684.8",
               "Churn %in% c('Yes','No')"
               ))
ve <- violatedEdits(E, telco)
summary(ve)
```

- Podsumowanie:

```
No violations detected, 11 checks evaluated to NA
```

```
TotalCharges
Min.   :  18.8
1st Qu.: 401.4
Median :1397.5
Mean   :2283.3
Qu.:3794.7
Max.   :8684.8
NA's   :11
```

- Summary(telco) wskazuje, że wszystkie NA znajdują sie w kolumnie TotalCharges, która posiada wartości numeryczne ciągłe. Wartości NA zostały zastąpione średnią z pozostałych wartości kolumny:

```
# Clean NA and replace with mean #
telco.clean <- telco
Mean <- mean(telco.clean[, 20], na.rm = TRUE)
telco.clean[, 20][is.na(telco.clean[, 20])] <- Mean
write.csv(telco.clean, file = "telco-clean.csv")
```

- Dla reszty wartości zostały zastosowane reguły zamieniające kategorie tekstowe na wartości numeryczne.

```
if (!is.na(gender) & gender == "Female") {
  gender <- 1
}
if (!is.na(gender) & gender == "Male") {
  gender <- 0
}
if (!is.na(Partner) & Partner == "Yes") {
  Partner <- 1
}
if (!is.na(Partner) & Partner == "No") {
  Partner <- 0
}
if (!is.na(Dependents) & Dependents == "Yes") {
  Dependents <- 1
}
if (!is.na(Dependents) & Dependents == "No") {
  Dependents <- 0
}
if (!is.na(PhoneService) & PhoneService == "Yes") {
  PhoneService <- 1
}
if (!is.na(PhoneService) & PhoneService == "No") {
  PhoneService <- 0
}
if (!is.na(MultipleLines) & MultipleLines == "Yes") {
  MultipleLines <- 1
}
if (!is.na(MultipleLines) & (MultipleLines == "No phone service" || MultipleLines == "No")) {
  MultipleLines <- 0
}
```

```
if (!is.na(InternetService) & InternetService == "No") {
  InternetService <- 0
}
if (!is.na(InternetService) & InternetService == "DSL") {
  InternetService <- 1
}
if (!is.na(InternetService) & InternetService == "Fiber optic") {
  InternetService <- 2
}
if (!is.na(OnlineSecurity) & (OnlineSecurity == "No internet service" | OnlineSecurity == "No")) {
  OnlineSecurity <- 0
}
if (!is.na(OnlineSecurity) & OnlineSecurity == "Yes") {
  OnlineSecurity <- 1
}
if (!is.na(OnlineBackup) & (OnlineBackup == "No internet service" | OnlineBackup == "No")) {
  OnlineBackup <- 0
}
if (!is.na(OnlineBackup) & OnlineBackup == "Yes") {
  OnlineBackup <- 1
}
if (!is.na(DeviceProtection) & (DeviceProtection == "No internet service" | DeviceProtection == "No")) {
  DeviceProtection <- 0
}
if (!is.na(DeviceProtection) & DeviceProtection == "Yes") {
  DeviceProtection <- 1
}
if (!is.na(TechSupport) & (TechSupport == "No internet service" | TechSupport == "No")) {
  TechSupport <- 0
}
if (!is.na(TechSupport) & TechSupport == "Yes") {
  TechSupport <- 1
}
if (!is.na(StreamingTV) & (StreamingTV == "No internet service" | StreamingTV == "No")) {
  StreamingTV <- 0
}
if (!is.na(StreamingTV) & StreamingTV == "Yes") {
  StreamingTV <- 1
}
if (!is.na(StreamingMovies) & (StreamingMovies == "No internet service" | StreamingMovies == "No")) {
  StreamingMovies <- 0
}
if (!is.na(StreamingMovies) & StreamingMovies == "Yes") {
  StreamingMovies <- 1
}
if (!is.na(Contract) & Contract == "Month-to-month") {
  Contract <- 1
}
if (!is.na(Contract) & Contract == "One year") {
  Contract <- 2
}
if (!is.na(Contract) & Contract == "Two year") {
  Contract <- 3
}
if (!is.na(PaperlessBilling) & PaperlessBilling == "No") {
  PaperlessBilling <- 0
}
if (!is.na(PaperlessBilling) & PaperlessBilling == "Yes") {
```

```
   PaperlessBilling <- 1
}
if (!is.na(PaymentMethod) & PaymentMethod == "Electronic check") {
   PaymentMethod <- 1
}
if (!is.na(PaymentMethod) & PaymentMethod == "Mailed check") {
   PaymentMethod <- 2
}
if (!is.na(PaymentMethod) & PaymentMethod == "Bank transfer (automatic)") {
   PaymentMethod <- 3
}
if (!is.na(PaymentMethod) & PaymentMethod == "Credit card (automatic)") {
   PaymentMethod <- 4
}
if (!is.na(Churn) & Churn == "Yes") {
   Churn <- 1
}
if (!is.na(Churn) & Churn == "No") {
   Churn <- 0
}
if (!is.na(tenure) & tenure <= 12) {
   tenure <- 1
}
if (!is.na(tenure) & tenure > 12 & tenure <= 24) {
   tenure <- 2
}
if (!is.na(tenure) & tenure > 24 & tenure <= 48) {
   tenure <- 4
}
if (!is.na(tenure) & tenure > 48 & tenure <= 60) {
   tenure <- 5
}
if (!is.na(tenure) & tenure > 60) {
   tenure <- 6
}
if (!is.na(MonthlyCharges) & MonthlyCharges <= 29) {
   MonthlyCharges <- 1
}
if (!is.na(MonthlyCharges) & MonthlyCharges > 29.6875 & MonthlyCharges <= 59.375) {
   MonthlyCharges <- 2
}
if (!is.na(MonthlyCharges) & MonthlyCharges > 59.375 & MonthlyCharges <= 89.0625) {
   MonthlyCharges <- 3
}
if (!is.na(MonthlyCharges) & MonthlyCharges > 89.0625) {
   MonthlyCharges <- 4
}
if (!is.na(TotalCharges) & TotalCharges <= 2171.2) {
   TotalCharges <- 1
}
if (!is.na(TotalCharges) & TotalCharges > 2171.2 & TotalCharges <= 4342.4) {
   TotalCharges <- 2
}
if (!is.na(TotalCharges) & TotalCharges > 4342.4 & TotalCharges <= 6513.6) {
   TotalCharges <- 3
}
if (!is.na(TotalCharges) & TotalCharges > 6513.6) {
   TotalCharges <- 4
```

```
}

# Correction rules #
rules <- correctionRules("rules2.txt")
corrected <- correctWithRules(rules, telco.clean)
telco.corrected <- corrected$corrected
```

- Po zastosowaniu reguł, kolumna customerID została usunięta, ponieważ jej wartości są unikalne.

```
# Remove customerID column and save corrected data #
telco.corrected$customerID <- NULL
write.csv(telco.corrected, file = "telco-corrected.csv")
telco.corrected <- read.csv("telco-corrected.csv", header=TRUE, sep=",")
telco.corrected$X <- NULL
```

- Wszystkie wartości zostały znormalizowane do wartości z przedziału 0-1.

```
# Normalize #
normalize <- function(x) {
  x <- as.numeric(x)
  num <- x - min(x)
  denom <- max(x) - min(x)
  return (num/denom)
}

telco.normalized <- telco.corrected
telco.normalized <- as.data.frame(lapply(telco.corrected[,1:20], normalize))
write.csv(telco.normalized, file = "telco-normalized.csv")

telco.normalized <- read.csv("telco-normalized.csv", header=TRUE, sep=",",
stringsAsFactors=FALSE)
telco.normalized$Churn = as.factor(telco.normalized$Churn)
telco.normalized$X <- NULL
```

- Dane zostały podzielone na zbiory testowe i treningowe:

Training: 67%
Test: 33%

```
# Training and test dataset #
set.seed(1234)
ind <- sample(2, nrow(telco.normalized), replace=TRUE, prob=c(0.67, 0.33))

# Compose training set
telco.training <- telco.normalized[ind==1, 1:20]

# Inspect training set
head(telco.training)

# Compose test set
```

```
telco.test <- telco.normalized[ind==2, 1:20]

# Inspect test set
head(telco.test)

# Compose training labels
telco.trainLabels <- telco.normalized[ind==1,20]

# Inspect result
print(telco.trainLabels)

# Compose test labels
telco.testLabels <- telco.normalized[ind==2, 20]

# Inspect result
print(telco.testLabels)
```

Klasyfikatory i ewaluacja

```
# Build the model
telco.KNNprediction <- knn(train = telco.training, test = telco.test, cl =
telco.trainLabels, k=2)

# Inspect
telco.KNNprediction

# Put in a data frame
telcoTestLabels <- data.frame(telco.testLabels)

# Merge pred and testLabels
telco.merge <- data.frame(telco.KNNprediction, telco.testLabels)

# Specify column names for `merge`
names(telco.merge) <- c("Predicted Churn", "Observed Churn")

# Inspect `merge`
telco.merge

correct <- function(merge) {
  count <- nrow(merge)
  rows <- c(1:count)
  result <- 0

  for(r in rows) {
    predicted <- as.character(merge[r, "Predicted Churn"])
    observed <- as.character(merge[r, "Observed Churn"])
    if(predicted == observed) {
      result <- result + 1
    }
  }
  return(cat("Correct predictions: ", result, "/", count, " = ", (re-
sult*100)/count, "%"))
```

```
}

correct(telco.merge)
# Correct predictions:  2253 / 2313  =  97.40597 %

KNNConfusion <- confusionMatrix(telco.KNNprediction,telco.test[,20])
KNNAccuracy <- KNNConfusion$overall[["Accuracy"]]

table(telco.KNNprediction)
confusionMatrix(telco.KNNprediction,telco.test[,20])
CrossTable(x = telco.testLabels, y = telco.KNNprediction, prop.chisq=FALSE)
```

KNN 1

```
telco.KNNprediction
   0    1
1711  602

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1669   42
         1   23  579

               Accuracy : 0.9719
                 95% CI : (0.9643, 0.9782)
    No Information Rate : 0.7315
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9278
 Mcnemar's Test P-Value : 0.02557

            Sensitivity : 0.9864
            Specificity : 0.9324
         Pos Pred Value : 0.9755
         Neg Pred Value : 0.9618
             Prevalence : 0.7315
         Detection Rate : 0.7216
   Detection Prevalence : 0.7397
      Balanced Accuracy : 0.9594

       'Positive' Class : 0



   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|
```

```
Total Observations in Table:  2313


                 | telco.KNNprediction
telco.testLabels |         0 |         1 | Row Total |
-----------------|-----------|-----------|-----------|
               0 |      1669 |        23 |      1692 |
                 |     0.986 |     0.014 |     0.732 |
                 |     0.975 |     0.038 |           |
                 |     0.722 |     0.010 |           |
-----------------|-----------|-----------|-----------|
               1 |        42 |       579 |       621 |
                 |     0.068 |     0.932 |     0.268 |
                 |     0.025 |     0.962 |           |
                 |     0.018 |     0.250 |           |
-----------------|-----------|-----------|-----------|
    Column Total |      1711 |       602 |      2313 |
                 |     0.740 |     0.260 |           |
-----------------|-----------|-----------|-----------|
```

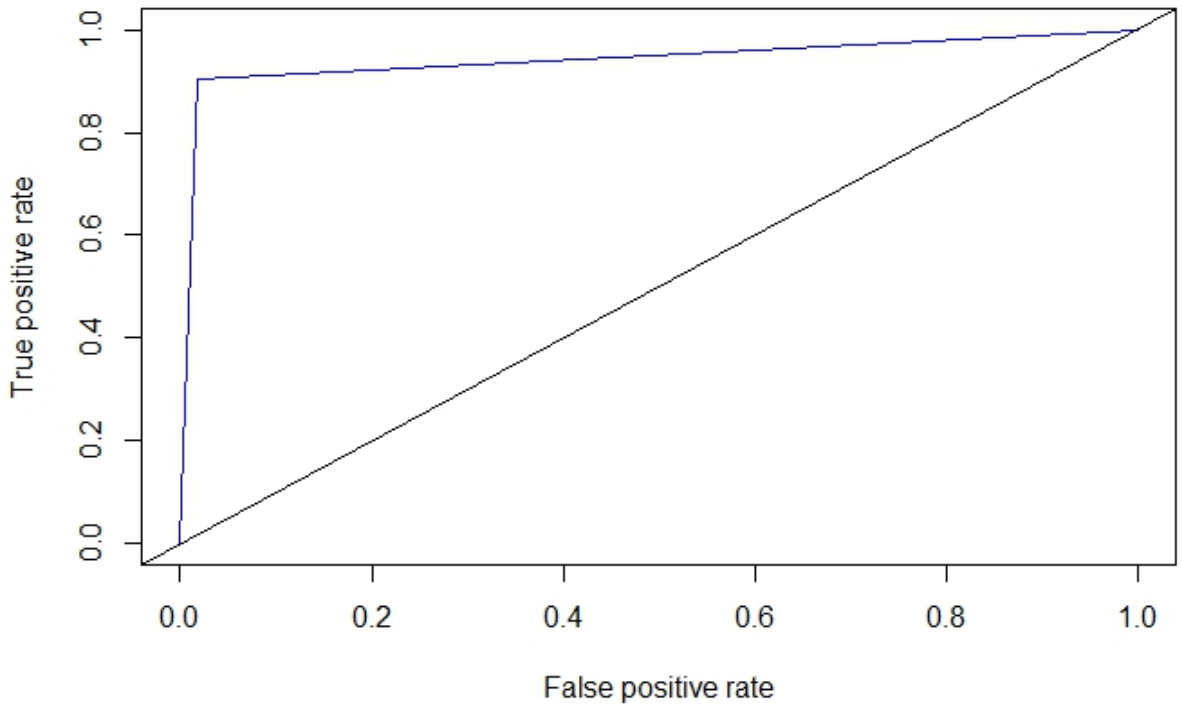
- **Ewaluacja**

```
#TPR = TP/TP + FN
#FPR = FP/FP + TN
TPR_KNN = KNNConfusion[["table"]][[2,2]] / (KNNConfusion[["table"]][[2,2]] +
KNNConfusion[["table"]][[1,2]])
FPR_KNN = KNNConfusion[["table"]][[2,1]] / (KNNConfusion[["table"]][[2,1]] +
KNNConfusion[["table"]][[1,1]])
TPR_KNN
FPR_KNN

pred <- prediction(as.numeric(telco.KNNprediction), as.numeric(tel-
co.test$Churn))
perf <- performance(pred,"tpr","fpr")
plot(perf,col="blue", type="l")
abline(0,1)

> TPR_KNN
[1] 0.9049919
> FPR_KNN
[1] 0.0177305
```

- **Macierz błędów**

```
          Reference
Prediction    0    1
         0 1669   42
         1   23  579
```

- **Krzywa ROC**



KNN 2 Caret

```
telco.KNN2prediction
   0    1
1811  502

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1490  321
         1  202  300

               Accuracy : 0.7739
                 95% CI : (0.7563, 0.7908)
    No Information Rate : 0.7315
    P-Value [Acc > NIR] : 1.631e-06

                  Kappa : 0.3872
 Mcnemar's Test P-Value : 2.472e-07
```

```
            Sensitivity : 0.8806
            Specificity : 0.4831
         Pos Pred Value : 0.8227
         Neg Pred Value : 0.5976
             Prevalence : 0.7315
         Detection Rate : 0.6442
   Detection Prevalence : 0.7830
      Balanced Accuracy : 0.6819

       'Positive' Class : 0
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|
```

```
Total Observations in Table:  2313
```

telco.testLabels	telco.KNN2prediction 0	1	Row Total
0	1490	202	1692
	0.881	0.119	0.732
	0.823	0.402	
	0.644	0.087	
1	321	300	621
	0.517	0.483	0.268
	0.177	0.598	
	0.139	0.130	
Column Total	1811	502	2313
	0.783	0.217	

Metoda bez użycia paczki Caret dawała lepsze rezultaty.

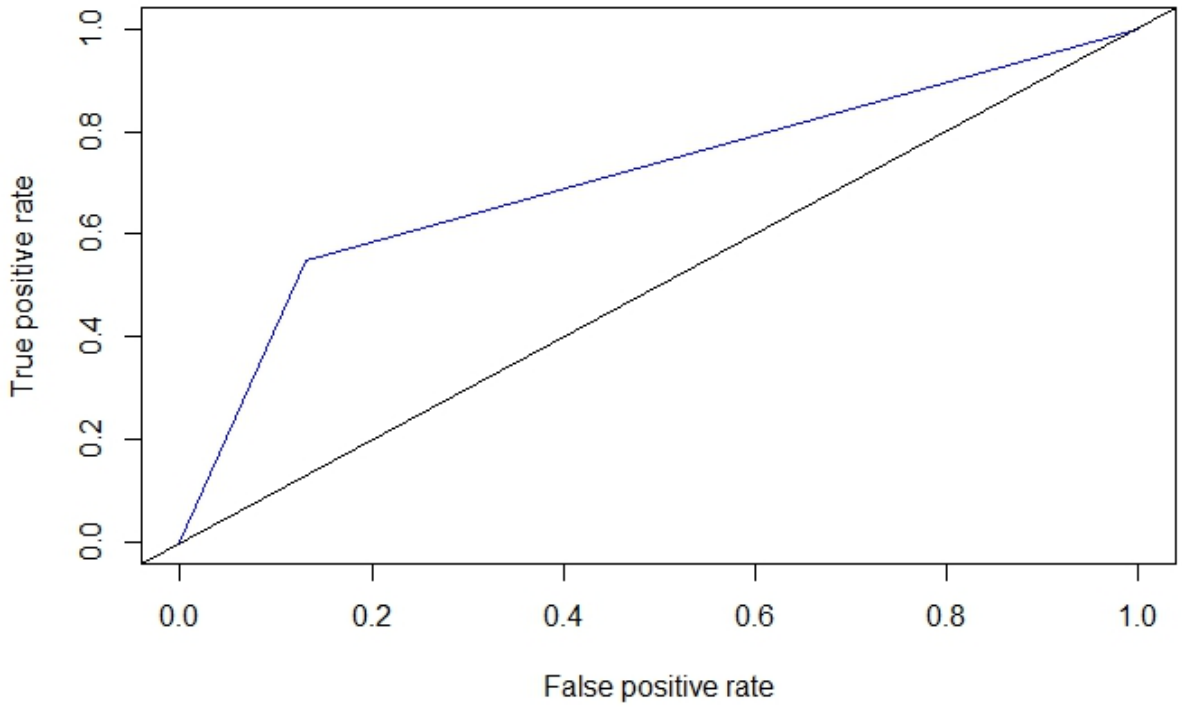
- **Ewaluacja**

```
> TPR_KNN2
[1] 0.5491143
> FPR_KNN2
[1] 0.1317967
```

- **Macierz błędów**

```
          Reference
Prediction    0    1
         0 1490  321
         1  202  300
```

- **Krzywa ROC**



Naive Bayes

```
telco.BAYESprediction
   0    1
1484  829

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1322  162
         1  370  459
```

```
               Accuracy : 0.77
                 95% CI : (0.7523, 0.787)
    No Information Rate : 0.7315
    P-Value [Acc > NIR] : 1.242e-05

                  Kappa : 0.4706
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7813
            Specificity : 0.7391
         Pos Pred Value : 0.8908
         Neg Pred Value : 0.5537
             Prevalence : 0.7315
         Detection Rate : 0.5716
   Detection Prevalence : 0.6416
      Balanced Accuracy : 0.7602

       'Positive' Class : 0
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|
```

```
Total Observations in Table:  2313
```

telco.testLabels	telco.BAYESprediction 0	1	Row Total
0	1322	370	1692
	0.781	0.219	0.732
	0.891	0.446	
	0.572	0.160	
1	162	459	621
	0.261	0.739	0.268
	0.109	0.554	
	0.070	0.198	
Column Total	1484	829	2313
	0.642	0.358	

- **Ewaluacja**

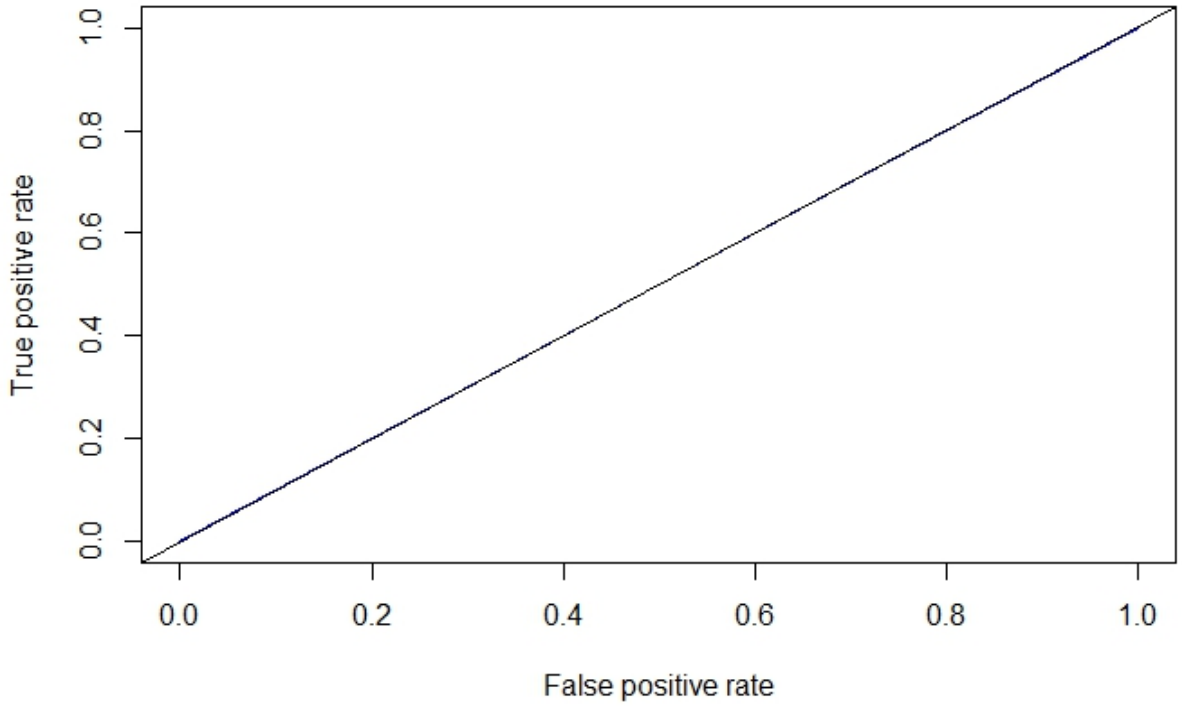
```
> TPR_bayes
[1] 0.7391304
```

```
> FPR_bayes
[1] 0.2192671
```

- **Macierz błędów**

```
          Reference
Prediction    0    1
         0 1322  162
         1  370  459
```

- **Krzywa ROC**



C4.5 Tree

```
telco.C45prediction
   0    1
1719  594

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1456  263
         1  236  358

               Accuracy : 0.7843
                 95% CI : (0.7669, 0.8009)
    No Information Rate : 0.7315
    P-Value [Acc > NIR] : 2.796e-09

                  Kappa : 0.4431
 Mcnemar's Test P-Value : 0.2445

            Sensitivity : 0.8605
            Specificity : 0.5765
         Pos Pred Value : 0.8470
         Neg Pred Value : 0.6027
             Prevalence : 0.7315
         Detection Rate : 0.6295
   Detection Prevalence : 0.7432
      Balanced Accuracy : 0.7185

       'Positive' Class : 0



   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  2313


                 | telco.C45prediction
telco.testLabels |         0 |         1 | Row Total |
-----------------|-----------|-----------|-----------|
               0 |      1456 |       236 |      1692 |
                 |     0.861 |     0.139 |     0.732 |
                 |     0.847 |     0.397 |           |
                 |     0.629 |     0.102 |           |
-----------------|-----------|-----------|-----------|
               1 |       263 |       358 |       621 |
                 |     0.424 |     0.576 |     0.268 |
                 |     0.153 |     0.603 |           |
                 |     0.114 |     0.155 |           |
```

```
-----------------|-----------|-----------|-----------|
    Column Total |      1719 |       594 |      2313 |
                 |     0.743 |     0.257 |           |
-----------------|-----------|-----------|-----------|
```

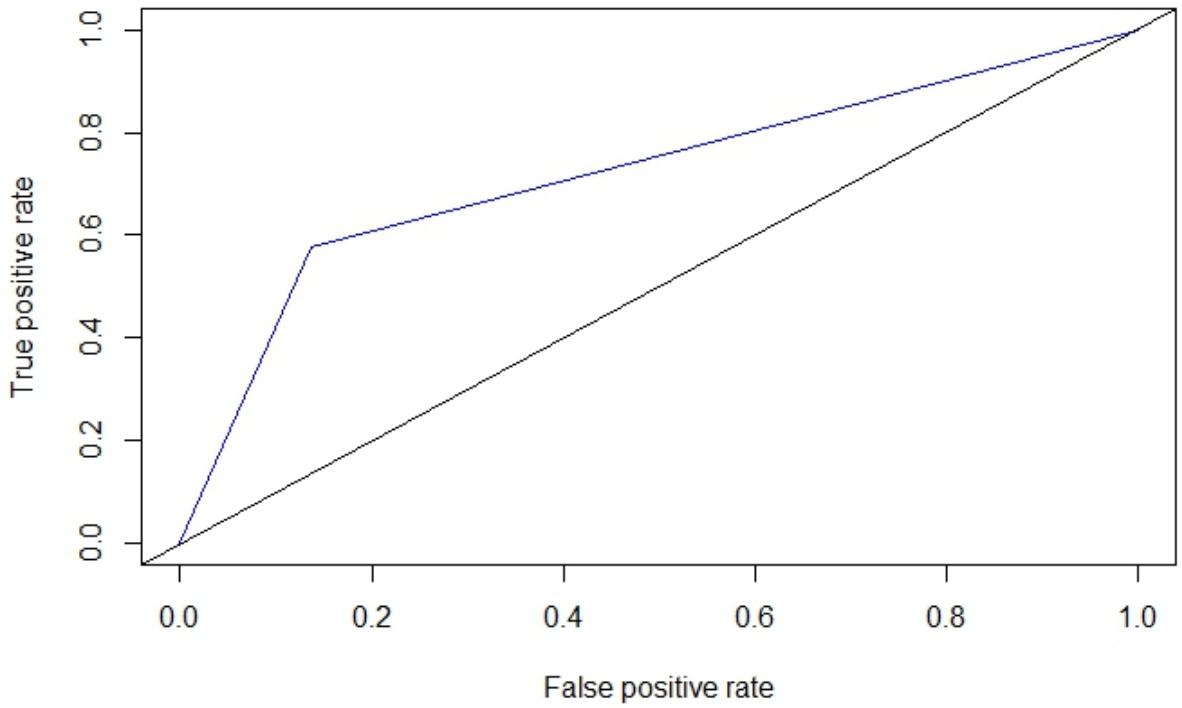
- **Ewaluacja**

```
> TPR_C45
[1] 0.5780998
> FPR_C45
[1] 0.1365248
```

- **Macierz błędów**

```
          Reference
Prediction    0    1
         0 1456  263
         1  236  358
```

- **Krzywa ROC**



Random Forest

```
telco.FORESTprediction
   0    1
1871  442

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1546  325
         1  146  296

               Accuracy : 0.7964
                 95% CI : (0.7794, 0.8126)
    No Information Rate : 0.7315
    P-Value [Acc > NIR] : 2.684e-13

                  Kappa : 0.4295
 Mcnemar's Test P-Value : 2.368e-16

            Sensitivity : 0.9137
            Specificity : 0.4767
         Pos Pred Value : 0.8263
         Neg Pred Value : 0.6697
             Prevalence : 0.7315
         Detection Rate : 0.6684
   Detection Prevalence : 0.8089
      Balanced Accuracy : 0.6952

       'Positive' Class : 0



   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  2313


                 | telco.FORESTprediction
telco.testLabels |         0 |         1 | Row Total |
-----------------|-----------|-----------|-----------|
               0 |      1546 |       146 |      1692 |
                 |     0.914 |     0.086 |     0.732 |
```

```
                 |     0.826 |     0.330 |           |
                 |     0.668 |     0.063 |           |
-----------------|-----------|-----------|-----------|
               1 |       325 |       296 |       621 |
                 |     0.523 |     0.477 |     0.268 |
                 |     0.174 |     0.670 |           |
                 |     0.141 |     0.128 |           |
-----------------|-----------|-----------|-----------|
    Column Total |      1871 |       442 |      2313 |
                 |     0.809 |     0.191 |           |
-----------------|-----------|-----------|-----------|
```

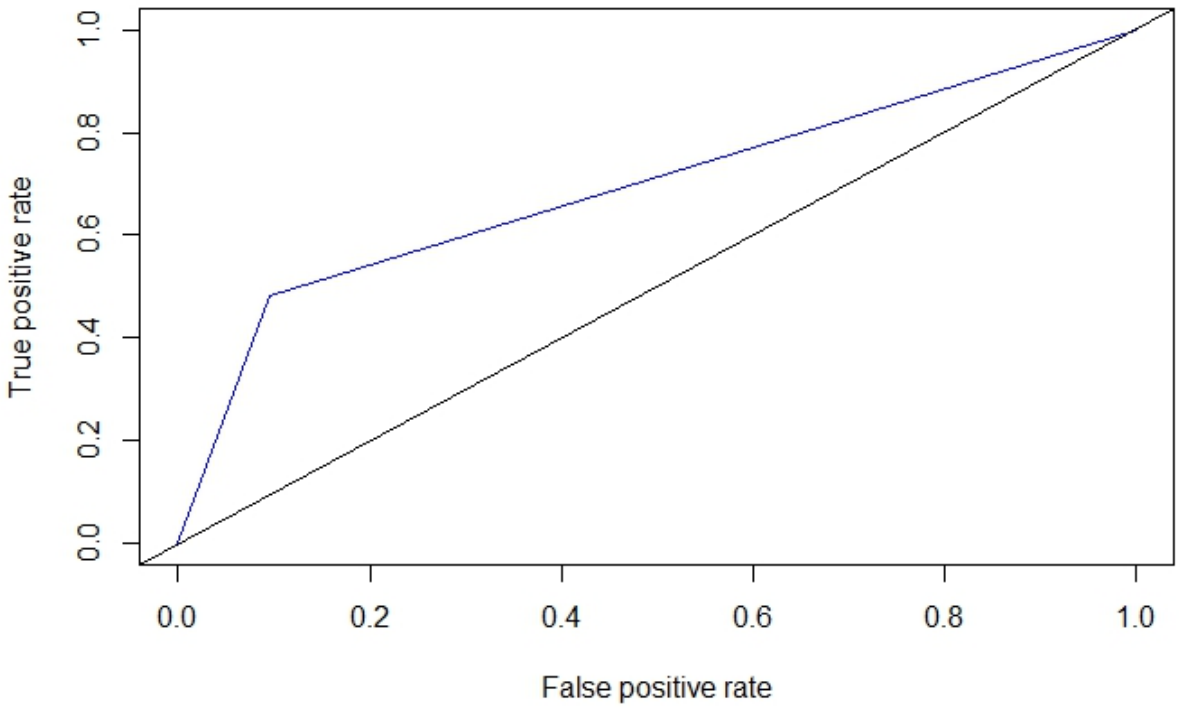
- **Ewaluacja**

```
> TPR_forest
[1] 0.4830918
> FPR_forest
[1] 0.09456265
```

- **Macierz błędów**

```
          Reference
Prediction    0    1
         0 1546  325
         1  146  296
```

- **Krzywa ROC**



Ewaluacja

Wartości:
TP - True Positive, klienci prawidłow zaklasyfikowani jako Rezygnacja - Tak
FP - False Positive, klienci nieprawidłowo zaklasyfikowani jako Rezygnacja - Tak
TN - True Negative, klienci prawidłowo zaklasyfikowani jako Rezygnacja - Nie
FN - False Negative, klienci nieprawidłowo zaklasyfikowani jako Rezygnacja - Nie

Przykład macierzy błędów(random forest):

```
          Reference
Prediction    0    1
         0 1546  325
         1  146  296
```

Prediction\Reference	0	1
0	TN	FN
1	FP	TP

Wzory:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

FNR = 1 - TPR

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$

TNR = 1 - FPR

Błędy:

Błąd pierwszego rodzaju: False Positive

Im więcej błędów pierwszego rodzaju, tym więcej odpowiedzi pozytywnych.

Dla bazy Telco jest to klient fałszywie zaklasyfikowany jako nierezygnujący z usług.

Błąd drugiego rodzaju: False Negative

Im więcej błędów drugiego rodzaju, tym więcej odpowiedzi pozytywnych.

Dla bazy Telco jest to klient fałszywie zaklasyfikowany jako rezygnujący z usług.

Dla bazy Telco gorsze będzie popełnienie błędu pierwszego rodzaju. Sytuacja odwzorowująca taki błąd, to klient fałszywie zaklasyfikowany jako nierezygnujący z usług, co może skutkować niewystarczającymi krokami podjętymi do utrzymania klienta i stratami.

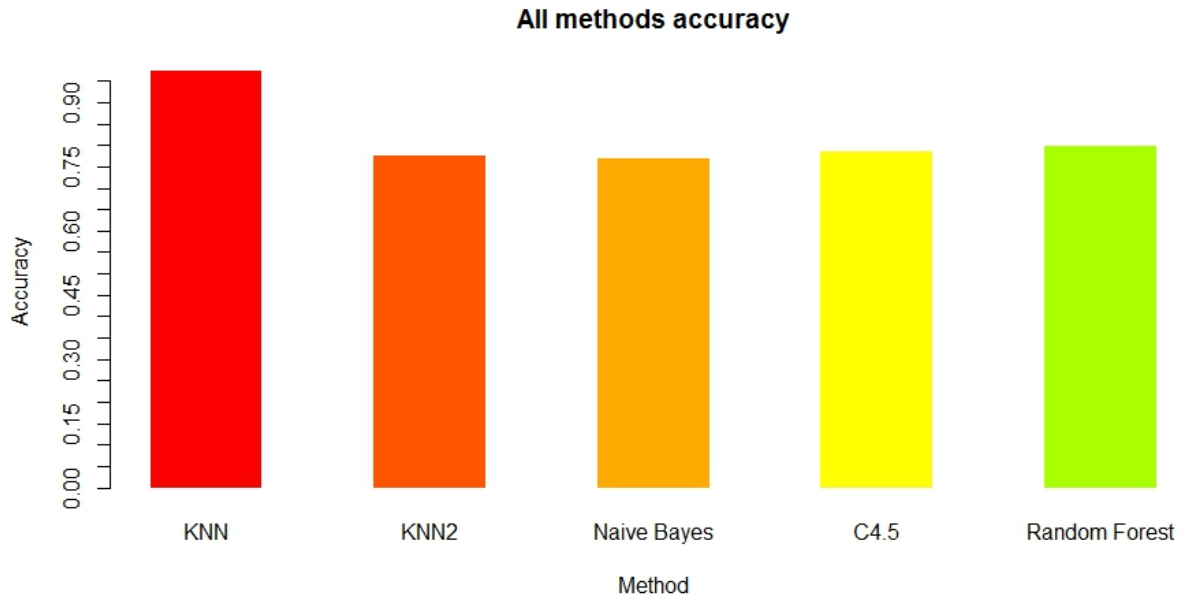
Krzywe ROC:

Idealna krzywa ROC przebiegałaby przez punkty 0,0->0,1->1,1, tworząc kąt prosty. Oznaczałoby to, że klasyfikacja przebiegła bezbłędnie.

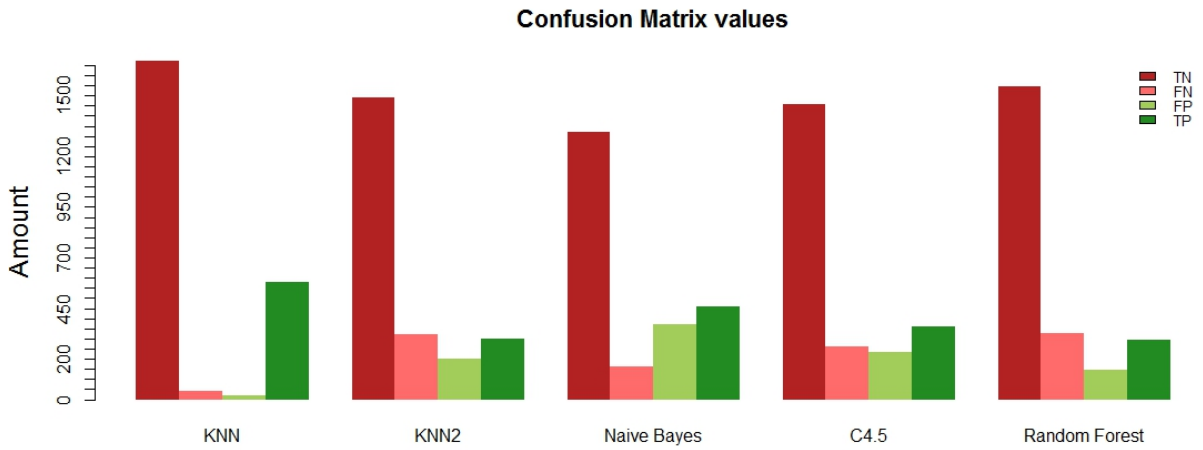
Najbliżej idealnego klasyfikatora znajduje się pierwsza metoda KNN i jednocześnie popełnia najmniej błędów pierwszego rodzaju, wybranych jako gorsze dla bazy Telco.

Wykresy

- **Wykres porównujący dokładność wszystkich metod:**



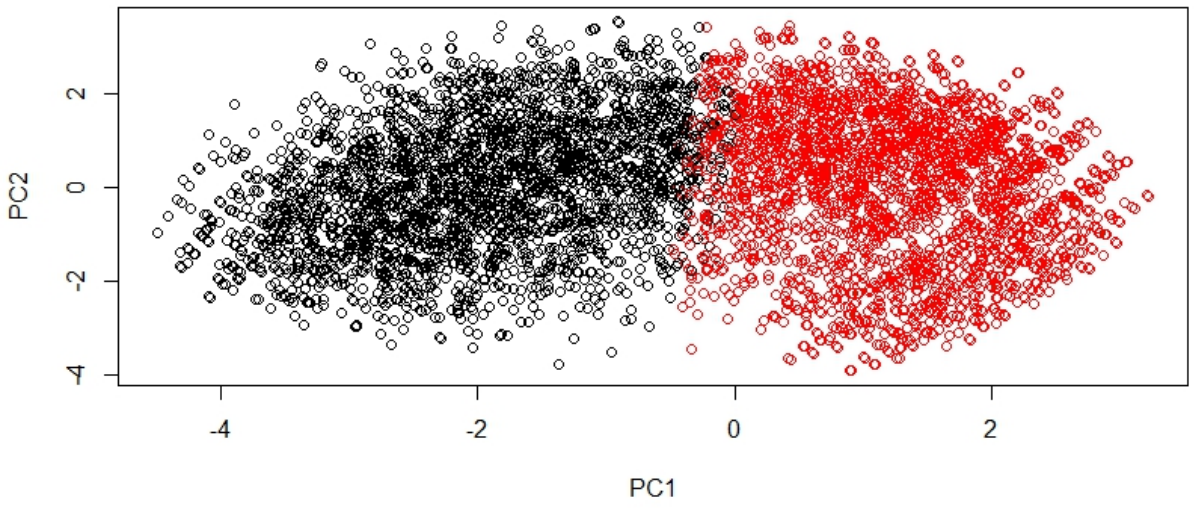
- **Wykres porównujący wartości TP, TN, FP, FN dla wszystkich metod:**



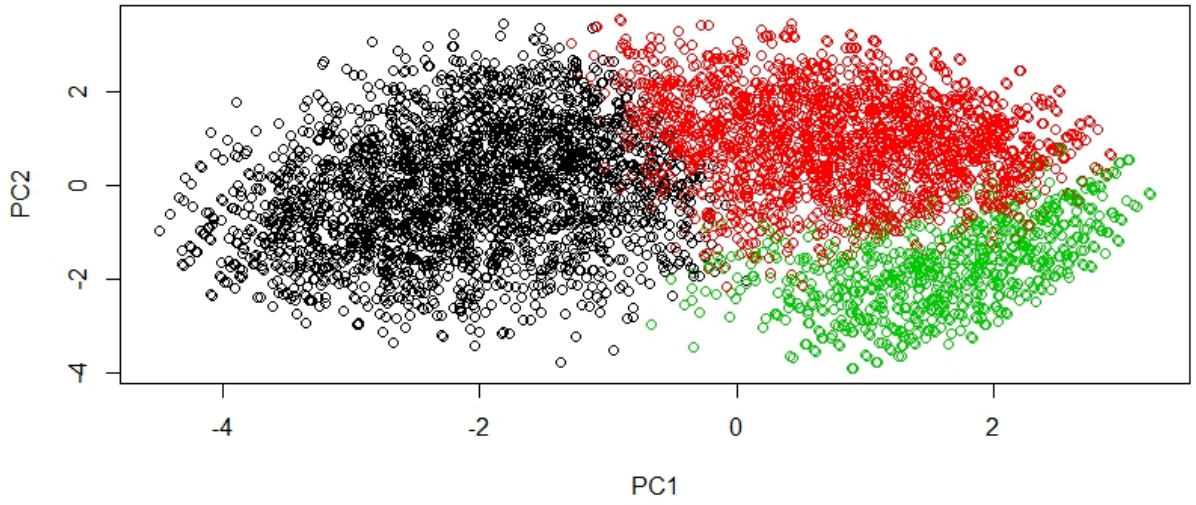
Najwięcej błędów wygenerowała klasyfikacja za pomocą NaiveBayes.

```
telco.scale <-  scale(telco.normalized[,-20], center=TRUE)
telco.pca <- prcomp(telco.scale)
telco.final <- predict(telco.pca)[,1:19]

set.seed(76964057) #Set the seed for reproducibility
k <-kmeans(telco.final, centers=3)
k$centers
table(k$cluster)

(cl <- kmeans(telco.final, 3))
plot(telco.final, col = cl$cluster)
points(cl$centers, col = 1:19, pch = 8, cex = 3)
```

- **K-średnich dla 2 klasyfikatorów:**



- **K-średnich dla 3 klasyfikatorów:**



Wymiary dla algorytmu k-średnich zostały stworzone z atrybutów o największym odchyleniu standardowym.

Reguły asocjacyjne

```
assocRules <- apriori(telco.correctedAssoc,
                      parameter = list(minlen=2, supp=0.2, conf=0.25),
                      appearance = list(rhs=c("Churn=1"),
                                        default="lhs"),
                      control = list(verbose=F))
```

Reguły asocjacyjne dla klientów rezygnujących:

```
   lhs                                rhs       support   confidence lift
[1] {OnlineSecurity=0,Contract=0} => {Churn=1} 0.2047423 0.4570523  1.722322
[2] {TechSupport=0,Contract=0}    => {Churn=1} 0.2057362 0.4522472  1.704215
[3] {PhoneService=1,Contract=0}   => {Churn=1} 0.2132614 0.4292655  1.617612
[4] {Contract=0}                  => {Churn=1} 0.2349851 0.4270968  1.609440
[5] {OnlineSecurity=0}            => {Churn=1} 0.2234843 0.3132962  1.180602
[6] {Dependents=0}                => {Churn=1} 0.2190828 0.3127914  1.178700
[7] {TechSupport=0}               => {Churn=1} 0.2213545 0.3118624  1.175199
[8] {TotalCharges=4}              => {Churn=1} 0.2028965 0.2978945  1.122563
[9] {PhoneService=1}              => {Churn=1} 0.2412324 0.2670964  1.006506
```

Najwyższe wartości confidence łączy reguła `Contract=0`, co może sugerować duży związek braku kontraktu z firmą, a rezygnacją klienta.

Kolejnymi ważnymi regułami są `OnlineSecurity=0 i TechSupport=0` w połączeniu

z `Contract=0`. Brak bezpieczeństwa w internecie mógł spowodować problemy i skłonić do rezygnacji klienta. Brak wsparcia technicznego również mógł utrudniać swobodne korzystanie z usług i spowodować rezygnację. Obie te usługi mogłyby zostać dołączone do pakietów z internetem, aby zapobiec rezygnacjom.

Podsumowanie

Dla bazy Telco pomocną metodą okazały się reguły asocjacyjne, ponieważ mogą wykazać przyczyny rezygnacji klientów z usług. Brak bezpieczeństwa w internecie mógł spowodować problemy i skłonić do rezygnacji klienta. Brak wsparcia technicznego również mógł utrudniać swobodne korzystanie z usług i spowodować rezygnację. Obie te usługi mogłyby zostać dołączone do pakietów z internetem, aby zapobiec rezygnacjom.

Random forest to jedna z metod klasyfikacji polagająca na tworzeniu wielu drzew decyzyjnych z losowego zestawu danych. Wyniki random forest nie różniły się znacznie od innych klasyfikatorów.

Dokładność wszystkich metod, poza pierwszą KNN, była zbliżona i wynosiła około 78%. Najbliżej idealnego klasyfikatora znajduje się pierwsza metoda KNN i jednocześnie popełnia najmniej błędów pierwszego rodzaju, wybranych jako gorsze dla bazy Telco.