Assignment 2 - Logistic Regression

Eoin Flynn

5 March 2018

Bond University
Data Science

Contents

Functions

This section will hold all of the functions that will be used throughout this markdown.

```
# Create a decision tree
createDecisionTreeModel <- function(formula, dataset, maxdepth) {
    suppressMessages(library(party))
    decisionTreeModel <- ctree(formula, data = dataset, controls = ctree_control(maxdepth = maxdepth))

    return(decisionTreeModel)
}

# Change rows to factors
setRowAsFactor <- function(dataset, columns) {
    for (column in columns) {
        dataset[, column] <- as.factor(dataset[, column])
    }
    return(dataset)
}
```

Data

In this section we will load in our data and do some basic data exploration

```
suppressMessages(library(RMySQL))

USER <- "root"
PASSWORD <- "A13337995"
HOST <- "localhost"
DBNAME <- "world"

statement <- "Select * from world.customerChurn"
db <- dbConnect(MySQL(), user = USER, password = PASSWORD, host = HOST,
    dbname = DBNAME, port = 3306)
customerDataset <- dbGetQuery(db, statement = statement)
dbDisconnect(db)
```

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

```
# Loops through and changes all relevant rows to factors and
# returns the dataset post modification


customerDataset <- setRowAsFactor(customerDataset, c("gender",
    "SeniorCitizen", "Partner", "Dependents", "PhoneService",
    "MultipleLines", "InternetService", "OnlineSecurity", "OnlineBackup",
    "DeviceProtection", "TechSupport", "StreamingTV", "StreamingMovies",
    "Contract", "PaperlessBilling", "PaymentMethod", "Churn"))

# Drop the columns that will not be needed
customerDataset = customerDataset[, -which(names(customerDataset) %in%
    c("customerID"))]
```

Split Data

We will now split our data into test and training sets. The purpose of this is to create a sample of data that the model has never seen before in order to gauge its accuracy. The training set will consist of 80% of the data while the remaining 20% will constitute the test set

```
suppressMessages(library(caTools))

# Set the seed to reproducability
set.seed(12216)

# Create our two datasets
sample <- sample.split(customerDataset, SplitRatio = 0.8)
train_df <- subset(customerDataset, sample == TRUE)
test_df <- subset(customerDataset, sample == FALSE)

# We can now see that the data is split approximately 80:20
print(sprintf("The full dataset has %s observations", NROW(customerDataset)))
```

```
## [1] "The full dataset has 7032 observations"
```

```
print(sprintf("The training dataset has %s observations", NROW(train_df)))
```

```
## [1] "The training dataset has 5628 observations"
```

```
print(sprintf("The testing dataset has %s observations", NROW(test_df)))
```

```
## [1] "The testing dataset has 1404 observations"
```

```
# Check to see how many customers churned in each dataset
table(train_df$Churn)
```

```
##
##   No  Yes
## 4120 1508
```

```
table(test_df$Churn)
```

```
##
##   No  Yes
## 1043  361
```

```
# We can see that each dataset holds approximately the same
# proportion of customers who churned
print(sprintf("%.2f%% of the training set churned", ((NROW(subset(train_df,
    Churn == "Yes")))/NROW(train_df) * 100)))
```

```
## [1] "26.79% of the training set churned"
```

```
print(sprintf("%.2f%% of the testing set churned", ((NROW(subset(test_df,
    Churn == "Yes")))/NROW(test_df) * 100)))
```

```
## [1] "25.71% of the testing set churned"
```

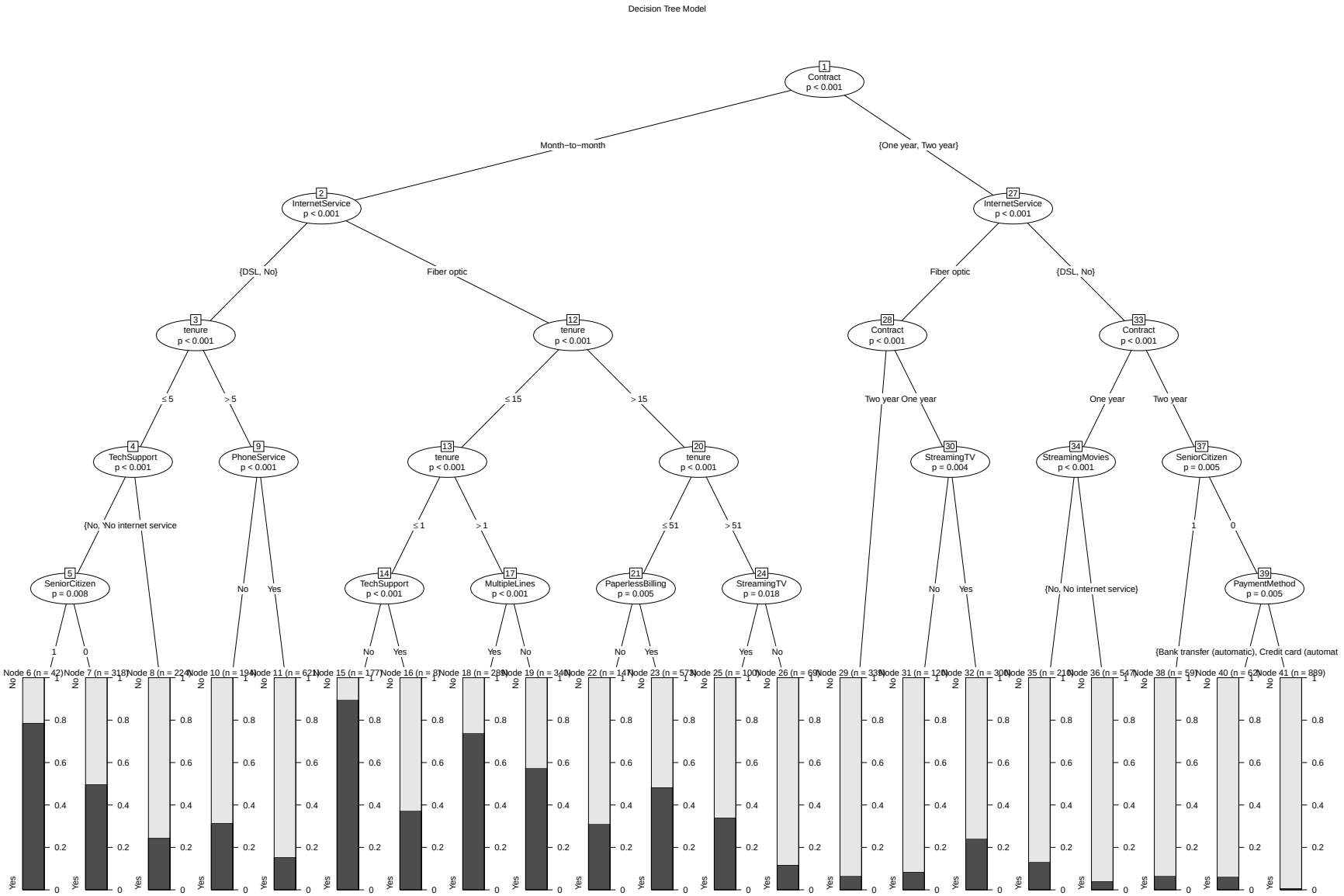
Decision Tree

We want to first make a decision tree to determine which variables are best able to predict whether a customer will churn. We already know from our previous report that the optimal model is however this time

the tree will only be run on the training dataset

```
# Create and plot the decision tree
decisionTreeModel = createDecisionTreeModel(formula = Churn ~
    ., dataset = train_df, maxdepth = 5)
```

```r
plot(decisionTreeModel, main = "Decision Tree Model", type = "extended",
    newpage = TRUE)
```


Decision Tree Model
1 Contract p < 0.001
Month−to−month
{One year, Two year}
2 InternetService p < 0.001
27 InternetService p < 0.001
{DSL, No}
Fiber optic
Fiber optic
{DSL, No}
3 tenure p < 0.001
12 tenure p < 0.001
28 Contract p < 0.001
33 Contract p < 0.001
≤ 5
> 5
≤ 15
> 15
Two year
One year
One year
Two year
4 TechSupport p < 0.001
9 PhoneService p < 0.001
13 tenure p < 0.001
20 tenure p < 0.001
30 StreamingTV p = 0.004
34 StreamingMovies p < 0.001
37 SeniorCitizen p = 0.005
{No, No internet service}
≤ 1
> 1
≤ 51
> 51
No
Yes
{No, No internet service}
1
0
5 SeniorCitizen p = 0.008
14 TechSupport p < 0.001
17 MultipleLines p < 0.001
21 PaperlessBilling p = 0.005
24 StreamingTV p = 0.018
39 PaymentMethod p = 0.005
1
0
No
Yes
No
Yes
Yes
No
No
Yes
Yes
No
{Bank transfer (automatic), Credit card (automat
Node 6 (n = 42)
Node 7 (n = 318)
Node 8 (n = 224)
Node 10 (n = 194)
Node 11 (n = 621)
Node 15 (n = 177)
Node 16 (n = 8)
Node 18 (n = 288)
Node 19 (n = 340)
Node 22 (n = 147)
Node 23 (n = 573)
Node 25 (n = 100)
Node 26 (n = 69)
Node 29 (n = 338)
Node 31 (n = 120)
Node 32 (n = 300)
Node 35 (n = 210)
Node 36 (n = 547)
Node 38 (n = 59)
Node 40 (n = 62)
Node 41 (n = 889)
No
Yes
0
0.2
0.4
0.6
0.8
1

From looking at the decision tree it is clear that the top three variables are Contract, InternetService, and Tenure. Below those three levels, other variables such as StreamingTV, and TechSupport become relevant. To develop the best performing model with this dataset we will create a regression using only those three top level variables, and then a second using all variables. The results from each model will then be compared before the best model is presented to management.