Today we will continue discussing customer lifecycle management. In particular, we concentrate on the retention problem. Business data analytics can help you identify who is about to churn by training classification model.

We will explore today a Telecommunication churn, and how we can detect it. Let’s proceed in our usual way:

```r
library(tidyverse)
library(data.table)
setwd(“path of your file”)
dt <- fread("edw_cdr.csv")
dt <- as.data.frame(dt)
dt <- dt[,c(6,1:5,7:ncol(dt))] # it is more convinient to have customer id in the first column
```

**Task 1 (1 p).** Let's investigate the dataset. What are the features that we need to keep, what to disregard, etc.

Now, we assign proper types to features. This is how we can change multiple features to factors:

```r
cols_to_factor <- c("customersuspended", "education", "gender", "homeowner", "churn", "maritalstatus", "occupation", "state", "usesinternetservice", "usesvoiceservice")
dt_cleaned <- dt %>%
  mutate_at(cols_to_factor, funs(factor(.)))
dt_cleaned <- dt_cleaned[!duplicated(dt_cleaned),]
#summary(dt_cleaned)
```

**Task 2 (1 p).** How many duplicates we deleted?

Next, let's split the dataset on a training and a test set (50/50).

**Task 3 (2 p).** Fill the blanks (hint: Check the previous lecture’s class assignment)

```r
set.seed(4698)
train_idx <- YOUR CODE HERE
train <- YOUR CODE HERE
test <- YOUR CODE HERE
str(dt_cleaned)
```
Let's first fit a very simple **logistic regression**:

```r
logistic_simple <- glm(data = train[, -1], formula = as.factor(churn) ~ calldroprate + gender, 
                        family = 'binomial')
summary(logistic_simple)
```

---

**Task 4 (2 p). Interpret above values !**

Let's fit it with all available data:

```r
logistic <- glm(data = train[, -1], formula = as.factor(churn) ~ ., family = 'binomial')
# we predict the churn porbability for a new test set and assing them to the test
# we predict the churn porbability for a new test set and assing them to the test
```

```r
test$logistic_predictions <- predict(newdata=test[, -1], logistic, type='response')
```
We have scores on a continuous scale from 0 to 1.

```
library(ggplot2)
ggplot(data = test, aes(x=logistic_predictions, fill=churn)) + geom_density(alpha=0.3) + theme_bw()
```

Now, we need to decide on a threshold. 0.5 is usually (but not always) a good choice.
### Decision trees

Let's briefly investigate, how to fit the single decision tree on a subset of data:

```r
library(rpart)
tree <- rpart(data=train[,,-1], churn ~ ., method='class')
#summary(tree)
library(rpart.plot)
rpart.plot(tree)
rpart.plot(tree, type=4, extra=101)
p <- predict(tree,test, type = "class")
table(test[,27],p) or table(test[,ncol(test)-2] ,p)
```

### Random Forest

```r
library(randomForest)
rfm <- randomForest(data = train, as.factor(churn)~.,)
#rfm
p <- predict(rfm,test)
table(test[,27],p)
mean(test[,27]==p) # Accuracy but optional
getTree(rf_m,500,labelVar = TRUE)
#importance(rfm)
#rf_m <- randomForest(data = train, as.factor(churn)~., importance = T, do.trace = F)
#test$rf_predictions <- predict(newdata=test, rf_m, type = 'prob')[,2]
```

### References:

1) [http://uc-r.github.io/logistic_regression](http://uc-r.github.io/logistic_regression)
3) [https://stats.idre.ucla.edu/r/dae/logit-regression/](https://stats.idre.ucla.edu/r/dae/logit-regression/)
7) [https://rstudio-pubs-static.s3.amazonaws.com/123908_3df6d25c8ad04030a2a429661125a59b.html#2](https://rstudio-pubs-static.s3.amazonaws.com/123908_3df6d25c8ad04030a2a429661125a59b.html#2) decision tree visualization