Case: Churn

Objective

Predict behavior to retain customers. You can analyze all relevant customer data and develop focused customer retention programs.

Preparation

- Set random seed
- Load libraries
- Set working directory
- Load data

```
set.seed(123)

library(ggplot2)
library(caret)
library(gbm)
library(rpart)
library(rpart.plot)

setwd("C:/Users/kfdek/Dropbox/Documents/R/Churn")

data <- read.csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

Data exploration

```
str(data)
```

```
## 'data.frame':    7043 obs. of  21 variables:
##  $ customerID      : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",..: 5376 3963 2565 5536 6512 6552 1003 4771 5605 4535 ...
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
```

```
##  $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
##  $ MultipleLines   : Factor w/ 3 levels "No","No phone service",..: 2 1 1 2 1 3 3 2 3 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2 2 2 1 2 1 ...
##  $ OnlineSecurity  : Factor w/ 3 levels "No","No internet service",..: 1 3 3 3 1 1 1 3 1 3 ...
##  $ OnlineBackup    : Factor w/ 3 levels "No","No internet service",..: 3 1 3 1 1 1 3 1 1 3 ...
##  $ DeviceProtection: Factor w/ 3 levels "No","No internet service",..: 1 3 1 3 1 3 1 1 3 1 ...
##  $ TechSupport     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ StreamingTV     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 3 1 3 1 ...
##  $ StreamingMovies : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 1 1 3 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1 1 1 1 2 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 4 4 1 3 3 2 4 3 1 ...
##  $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...
```

```r
summary(data)
```

```
##       customerID      gender     SeniorCitizen     Partner    Dependents
##  0002-ORFBO:   1   Female:3488   Min.   :0.0000   No :3641   No :4933
##  0003-MKNFE:   1   Male  :3555   1st Qu.:0.0000   Yes:3402   Yes:2110
##  0004-TLHLJ:   1                 Median :0.0000
##  0011-IGKFF:   1                 Mean   :0.1621
##  0013-EXCHZ:   1                 3rd Qu.:0.0000
##  0013-MHZWF:   1                 Max.   :1.0000
##  (Other)   :7037
##      tenure      PhoneService          MultipleLines     InternetService
##  Min.   : 0.00   No : 682     No              :3390   DSL        :2421
##  1st Qu.: 9.00   Yes:6361     No phone service: 682   Fiber optic:3096
##  Median :29.00                Yes             :2971   No         :1526
##  Mean   :32.37
##  3rd Qu.:55.00
##  Max.   :72.00
##
```

```
##             OnlineSecurity              OnlineBackup
##  No                 :3498   No                 :3088
##  No internet service:1526   No internet service:1526
##  Yes                :2019   Yes                :2429
##
##
##
##
##            DeviceProtection              TechSupport
##  No                 :3095    No                 :3473
##  No internet service:1526    No internet service:1526
##  Yes                :2422    Yes                :2044
##
##
##
##
##              StreamingTV             StreamingMovies
##  No                 :2810   No                 :2785
##  No internet service:1526   No internet service:1526
##  Yes                :2707   Yes                :2732
##
##
##
##
##           Contract    PaperlessBilling                   PaymentMethod
##  Month-to-month:3875   No :2872         Bank transfer (automatic):1544
##  One year      :1473   Yes:4171         Credit card (automatic)  :1522
##  Two year      :1695                    Electronic check         :2365
##                                         Mailed check             :1612
##
##
##
##  MonthlyCharges    TotalCharges    Churn
##  Min.   : 18.25   Min.   :  18.8   No :5174
##  1st Qu.: 35.50   1st Qu.: 401.4   Yes:1869
##  Median : 70.35   Median :1397.5
##  Mean   : 64.76   Mean   :2283.3
##  3rd Qu.: 89.85   3rd Qu.:3794.7
##  Max.   :118.75   Max.   :8684.8
```

```
##                    NA's   :11
```

```
prop.table(table(data$Churn))
```

```
##
##        No       Yes
## 0.7346301 0.2653699
```

Data clean-up

- Remove customerID
- Convert SeniorCitizen to factor variable

```
data <- data[, !colnames(data) == "customerID"]
data$SeniorCitizen <- factor(data$SeniorCitizen)
```

Missing data

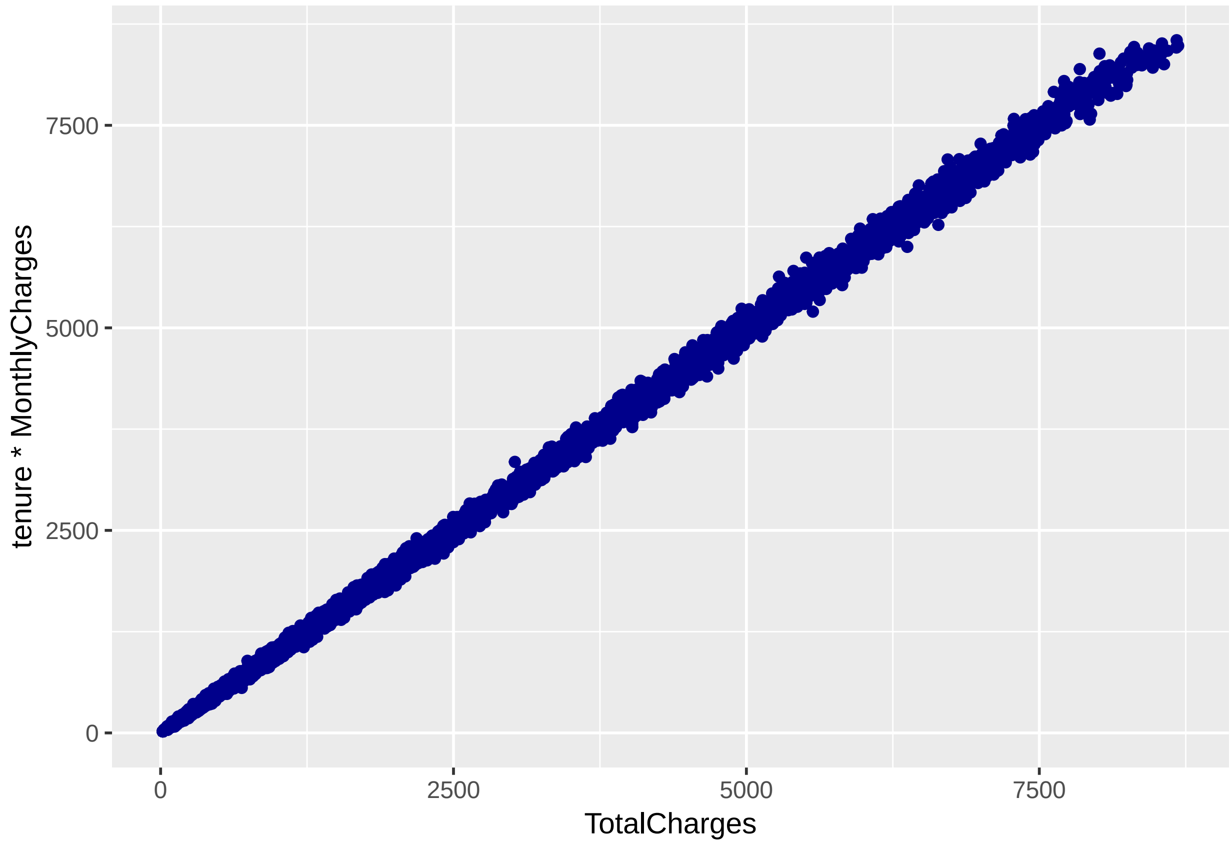
- Usual options: Impute missing values, remove rows with missing values, use model that can handle missing values
- Alternative: Remove feature TotalCharges which has all the missing data, because it is tenure * MonthlyCharges

```
round(cor(data$tenure * data$MonthlyCharges, data$TotalCharges,
          use = "pairwise.complete.obs"), 3)
```

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

```
ggplot(data, aes(x = TotalCharges,  y = tenure * MonthlyCharges)) +
  geom_point(color = "blue4")
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```




```
data <- data[, !colnames(data) == "TotalCharges"]
```

Sanity Checks

- Calculate some tables of features with expected overlap
- Plot some expected relationships

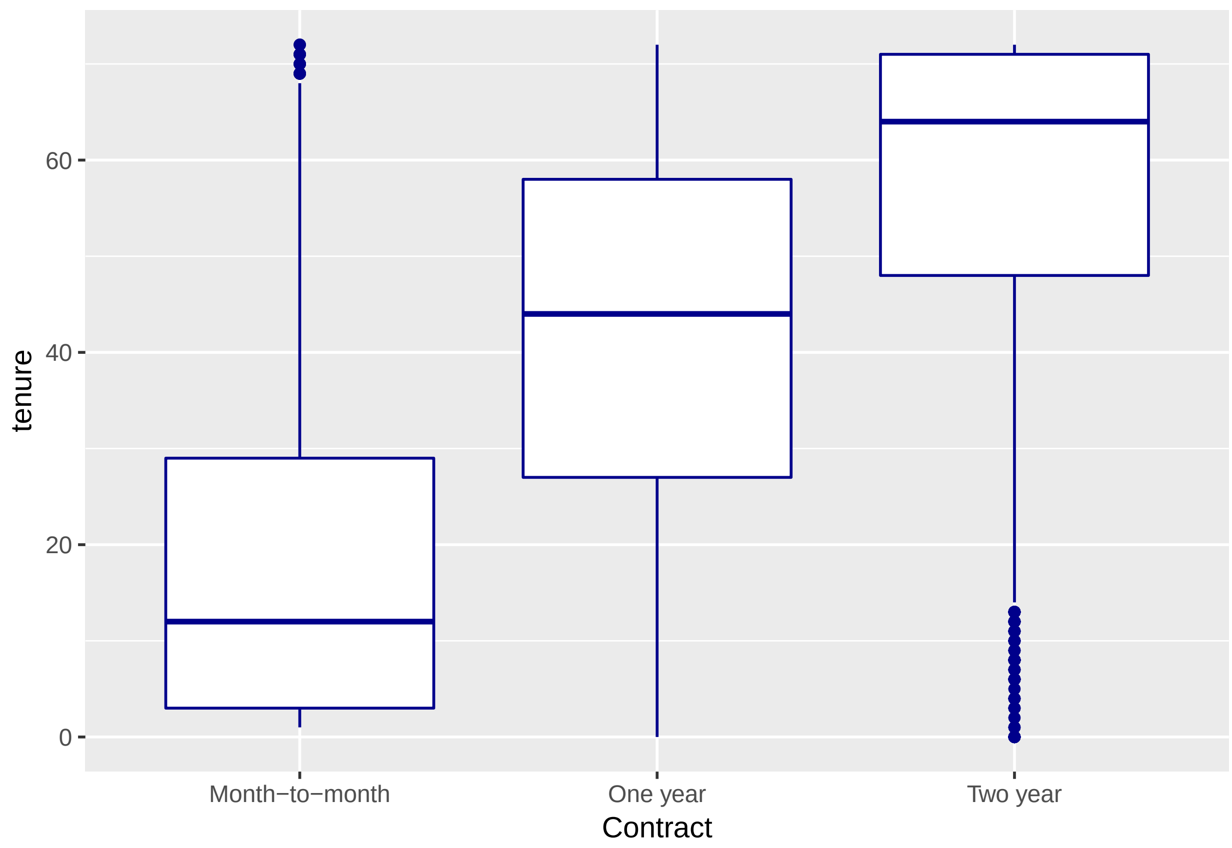
```
with(data, table(InternetService, OnlineSecurity))
```

```
##                  OnlineSecurity
## InternetService   No No internet service  Yes
##     DSL         1241                   0 1180
##     Fiber optic 2257                   0  839
##     No             0                1526    0
```

```r
with(data, table(PhoneService, MultipleLines))
```

```
##             MultipleLines
## PhoneService   No No phone service  Yes
##          No     0              682    0
##          Yes 3390                0 2971
```

```r
ggplot(data, aes(x = Contract, y = tenure)) + geom_boxplot(color = "blue4")
```



Analysis plan

Preprocessing

- Scale data (mean = 0, sd = 1) to ensure features with high range of values do not dominate

Model parameters

- Use decision tree model
  - Has integrated feature selection
  - Tree provides insights in how selected features determine churn rate
- Use classification accuracy instead of ROC/AUC as training metric
  - Labels of Churn are fairly balanced
  - No preference for rate of true positives and false positives
- Use 10 fold cross-validation

Evaluation

- Compare test set accuracy and feature importance with logistic regression baseline model and strong gradient boosting model

Run models

Split data in train and test sets using balanced split

```
split <- createDataPartition(data$Churn, p = 0.3, list = F)
train <- data[-split, ]
test <- data[split, ]
```

Baseline logistic regression model

```
fit <- train(Churn ~ ., data = train, method = "glm", preProcess = c("center", "scale"),
             trControl = trainControl(method = "cv"))
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
```

```
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
paste("Accuracy:", round(sum(test$Churn == predict(fit, test)) / nrow(test), 3))
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## [1] "Accuracy: 0.799"
```

```
varImp(fit)
```

```
## glm variable importance
##
```

```
##   only 20 most important variables shown (out of 22)
##
##                                       Overall
## tenure                                100.000
## `ContractTwo year`                     60.870
## `ContractOne year`                     52.109
## `PaymentMethodElectronic check`        27.794
## PaperlessBillingYes                    27.180
## MultipleLinesYes                       24.094
## SeniorCitizen1                         19.430
## `InternetServiceFiber optic`           18.222
## InternetServiceNo                      18.211
## genderMale                             16.062
## StreamingMoviesYes                     15.104
## StreamingTVYes                         14.201
## OnlineSecurityYes                      13.384
## DependentsYes                          10.035
## MonthlyCharges                          9.801
## `PaymentMethodCredit card (automatic)`  9.564
## DeviceProtectionYes                     6.783
## TechSupportYes                          6.335
## `PaymentMethodMailed check`             3.949
## PhoneServiceYes                         3.897
```

Baseline logistic regression model 2

- Remove correlated features to fix rank-deficiency warning

```
fit <- train(Churn ~ ., data = train, method = "glm", preProcess = c("center", "scale", "corr"),
             trControl = trainControl(method = "cv"))
paste("Accuracy:", round(sum(test$Churn == predict(fit, test)) / nrow(test), 3))
```

```
## [1] "Accuracy: 0.799"
```

```
varImp(fit)
```

```
## glm variable importance
```

```
##
##   only 20 most important variables shown (out of 22)
##
##                                         Overall
## tenure                                  100.000
## `ContractTwo year`                       60.870
## `ContractOne year`                       52.109
## `PaymentMethodElectronic check`          27.794
## PaperlessBillingYes                      27.180
## MultipleLinesYes                         24.094
## SeniorCitizen1                           19.430
## `InternetServiceFiber optic`             18.222
## `StreamingMoviesNo internet service`     18.211
## genderMale                               16.062
## StreamingMoviesYes                       15.104
## StreamingTVYes                           14.201
## OnlineSecurityYes                        13.384
## DependentsYes                            10.035
## MonthlyCharges                            9.801
## `PaymentMethodCredit card (automatic)`    9.564
## DeviceProtectionYes                       6.783
## TechSupportYes                            6.335
## `PaymentMethodMailed check`               3.949
## PhoneServiceYes                           3.897
```

Strong gradient boosting model

```r
fit <- train(Churn ~ ., data = train, method = "gbm", preProcess = c("center", "scale"),
             trControl = trainControl(method = "cv"), verbose = F)
paste("Accuracy:", round(sum(test$Churn == predict(fit, test)) / nrow(test), 3))
```

```
## [1] "Accuracy: 0.807"
```

```r
varImp(fit)
```

```
## gbm variable importance
```

```
##
##   only 20 most important variables shown (out of 29)
##
##                                      Overall
## tenure                              100.0000
## InternetServiceFiber optic           59.7776
## PaymentMethodElectronic check        35.6634
## ContractTwo year                     26.2592
## InternetServiceNo                     8.4449
## ContractOne year                      8.3648
## MonthlyCharges                        6.5582
## OnlineSecurityYes                     5.6344
## SeniorCitizen1                        4.7177
## PaperlessBillingYes                   4.0332
## MultipleLinesYes                      2.8625
## TechSupportYes                        2.4289
## StreamingMoviesYes                    1.8413
## MultipleLinesNo phone service         1.1586
## StreamingTVYes                        0.8876
## OnlineBackupYes                       0.8826
## PhoneServiceYes                       0.7850
## DependentsYes                         0.3958
## genderMale                            0.3858
## PaymentMethodCredit card (automatic)   0.0000
```

Decision tree model

```
fit <- train(Churn ~ ., data = train, method = "rpart", preProcess = c("center", "scale"),
             trControl = trainControl(method = "cv"))
paste("Accuracy:", round(sum(test$Churn == predict(fit, test)) / nrow(test), 3))
```

```
## [1] "Accuracy: 0.78"
```

```
varImp(fit)
```

```
## rpart variable importance
```

```
##
##   only 20 most important variables shown (out of 33)
##
##                                         Overall
## tenure                                  100.000
## InternetServiceFiber optic               97.668
## ContractTwo year                         78.365
## PaymentMethodElectronic check            70.731
## InternetServiceNo                        43.085
## ContractOne year                         21.578
## OnlineSecurityYes                        13.905
## MonthlyCharges                            9.074
## `OnlineBackupNo internet service`         0.000
## `InternetServiceFiber optic`              0.000
## PaperlessBillingYes                       0.000
## `PaymentMethodCredit card (automatic)`    0.000
## `MultipleLinesNo phone service`           0.000
## SeniorCitizen1                            0.000
## TechSupportYes                            0.000
## PartnerYes                                0.000
## `PaymentMethodMailed check`               0.000
## DependentsYes                             0.000
## genderMale                                0.000
## OnlineBackupYes                           0.000
```

Decision tree model 2

- Remove scaling to get non-scaled rules
    - For models where scaling is important an alternative is to scale predictions back to original mean and sd instead of removing preprocessing

```r
fit <- train(Churn ~ ., data = train, method = "rpart", trControl = trainControl(method = "cv"))
paste("Accuracy:", round(sum(test$Churn == predict(fit, test)) / nrow(test), 3))
```

```
## [1] "Accuracy: 0.78"
```

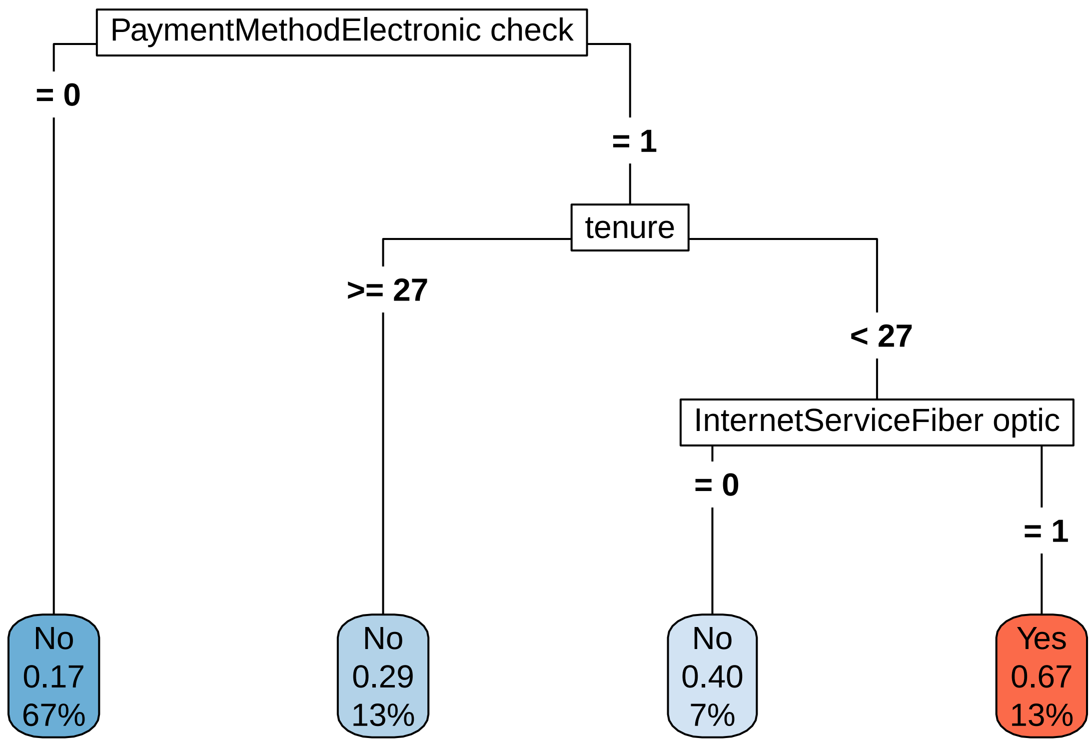
```r
varImp(fit)
```

```
## rpart variable importance
##
##   only 20 most important variables shown (out of 33)
##
##                                       Overall
## tenure                                100.000
## InternetServiceFiber optic             97.668
## ContractTwo year                       78.365
## PaymentMethodElectronic check          70.731
## InternetServiceNo                      43.085
## ContractOne year                       21.578
## OnlineSecurityYes                      13.905
## MonthlyCharges                          9.074
## TechSupportYes                          0.000
## PaperlessBillingYes                     0.000
## StreamingTVYes                          0.000
## PhoneServiceYes                         0.000
## `TechSupportNo internet service`        0.000
## `StreamingMoviesNo internet service`    0.000
## `InternetServiceFiber optic`            0.000
## `ContractOne year`                      0.000
## `PaymentMethodMailed check`             0.000
## `DeviceProtectionNo internet service`   0.000
## `OnlineSecurityNo internet service`     0.000
## `OnlineBackupNo internet service`       0.000
```

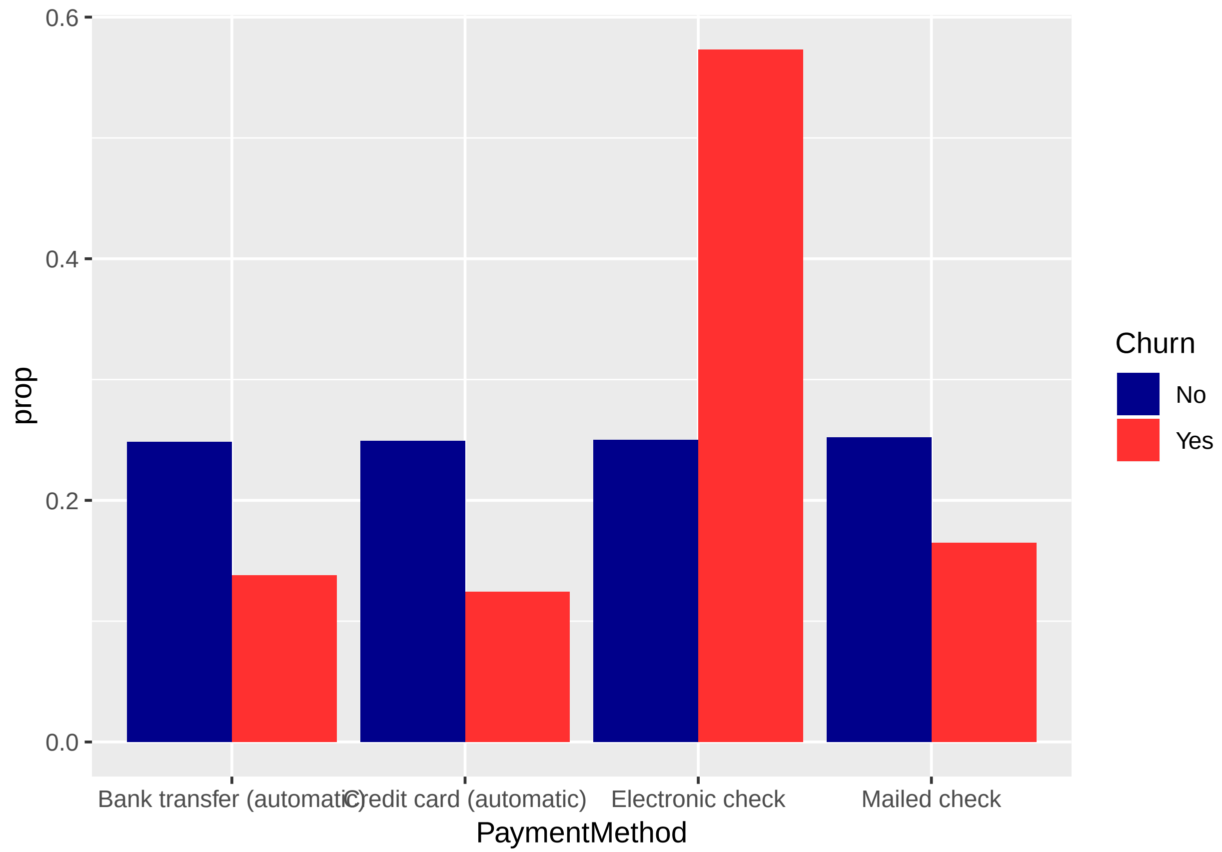
Data visualization

- Plot tree
- Plot relationships between Churn and features

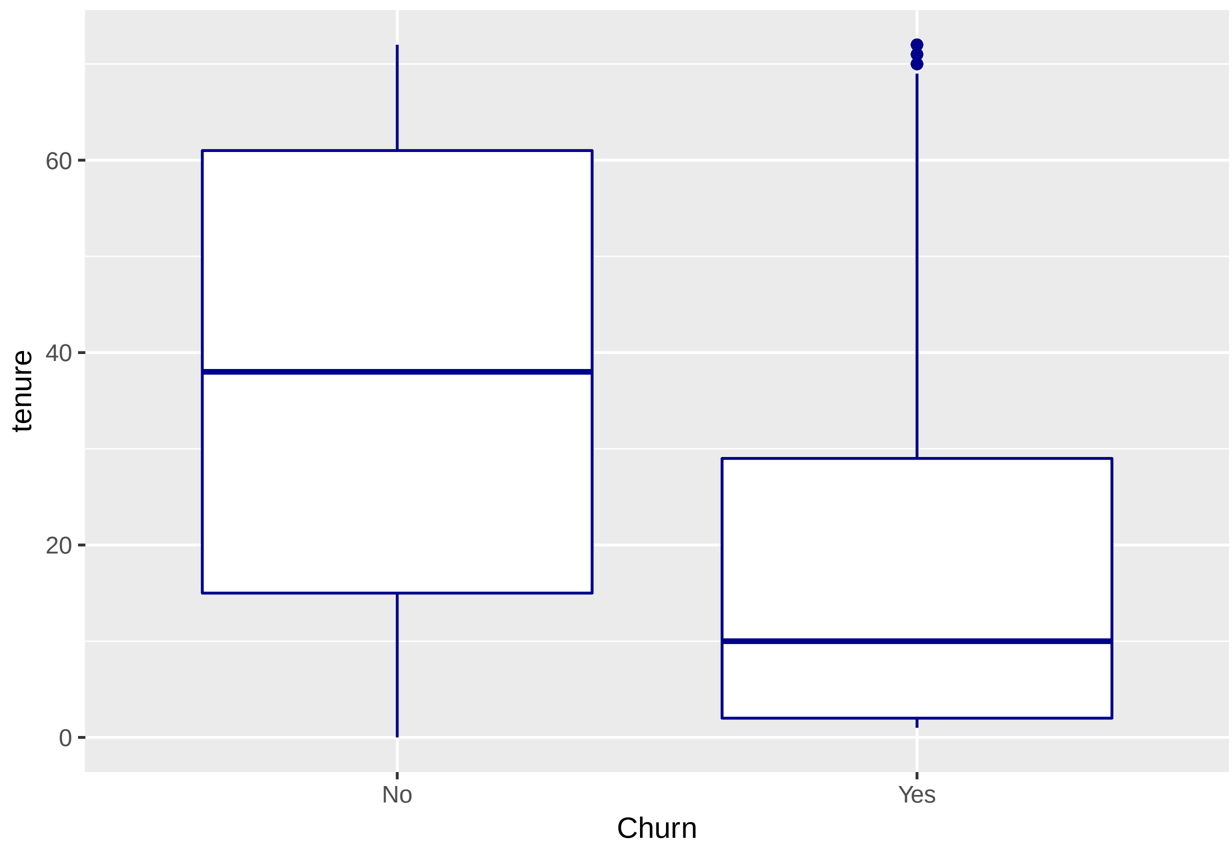
```r
rpart.plot(fit$finalModel, type = 5, cex = 1, box.palette = "BuRd")
```



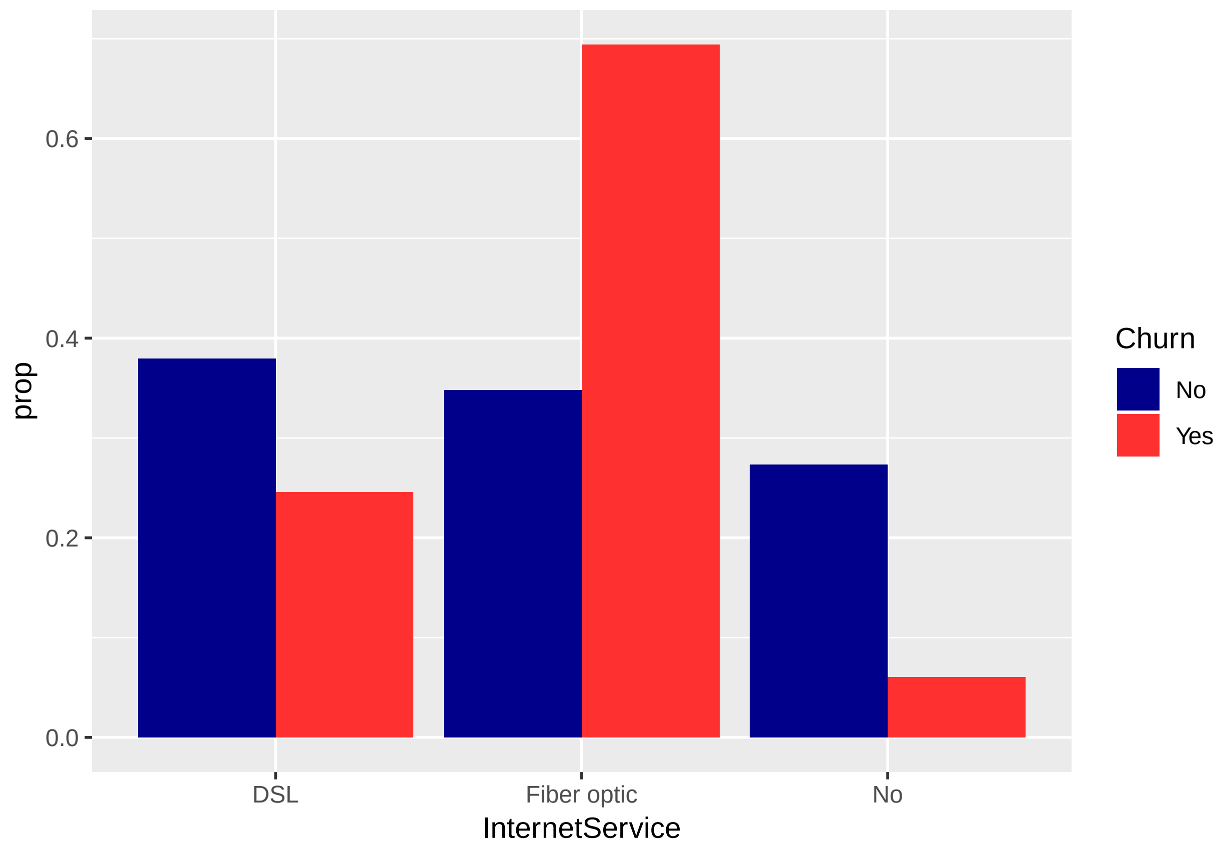

```r
ggplot(data, aes(x = PaymentMethod, y = ..prop.., group = Churn, fill = Churn)) +
  geom_bar(position = "dodge") + scale_fill_manual(values = c("blue4", "firebrick1"))
```




```
ggplot(data, aes(x = Churn, y = tenure)) + geom_boxplot(color = "blue4")
```




```r
ggplot(data, aes(x = InternetService, y = ..prop.., group = Churn, fill = Churn)) +
  geom_bar(position = "dodge") + scale_fill_manual(values = c("blue4", "firebrick1"))
```



Save data

```r
save(data, fit, test, file = "data.RData")
```

Customer retention strategies

- Target customers that pay with electronic checks
- Target customers that have a tenure < 27 that pay with electronic checks

- Target customers that have fiber optic internet service, a tenure < 27 and pay with electronic checks

Future improvements

- Ask data generators/curators if SeniorCitizen label 1 corresponds to customer being a senior citizen
- Ask data generators/curators why TotalCharges differs a little bit from tenure * MonthlyCharges
- Improve decision tree
  - evaluate bootstrapping instead of cross-validation
  - evaluate parameters for splitting
  - evaluate parameters for pruning
- Improve visualizations and app
- Secondary objective: Improve prediction of customers that might churn at the cost of minor increase in false positives (and decreased accuracy) using ROC/AUC

Print session info

```r
sessionInfo()
```

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows >= 8 x64 (build 9200)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] parallel  splines   stats     graphics  grDevices utils     datasets
## [8] methods   base
##
```

```
## other attached packages:
## [1] rpart.plot_3.0.4 rpart_4.1-13     gbm_2.1.3        survival_2.42-3
## [5] caret_6.0-80     lattice_0.20-35  ggplot2_3.0.0    rmarkdown_1.10
##
## loaded via a namespace (and not attached):
##  [1] magic_1.5-8        ddalpha_1.3.4      tidyr_0.8.1
##  [4] sfsmisc_1.1-2      foreach_1.4.4      prodlim_2018.04.18
##  [7] assertthat_0.2.0   stats4_3.5.1       DRR_0.0.3
## [10] yaml_2.2.0         robustbase_0.93-2  ipred_0.9-7
## [13] pillar_1.3.0       backports_1.1.2    glue_1.3.0
## [16] digest_0.6.15      colorspace_1.3-2   recipes_0.1.3
## [19] htmltools_0.3.6    Matrix_1.2-14      plyr_1.8.4
## [22] timeDate_3043.102  pkgconfig_2.0.2    CVST_0.2-2
## [25] broom_0.5.0        purrr_0.2.5        scales_1.0.0
## [28] gower_0.1.2        lava_1.6.3         tibble_1.4.2
## [31] withr_2.1.2        nnet_7.3-12        lazyeval_0.2.1
## [34] magrittr_1.5       crayon_1.3.4       evaluate_0.11
## [37] nlme_3.1-137       MASS_7.3-50        dimRed_0.1.0
## [40] class_7.3-14       tools_3.5.1        data.table_1.11.4
## [43] stringr_1.3.1      kernlab_0.9-27     munsell_0.5.0
## [46] bindrcpp_0.2.2     pls_2.7-0          compiler_3.5.1
## [49] e1071_1.7-0        RcppRoll_0.3.0     rlang_0.2.2
## [52] grid_3.5.1         iterators_1.0.10   labeling_0.3
## [55] geometry_0.3-6     gtable_0.2.0       ModelMetrics_1.2.0
## [58] codetools_0.2-15   abind_1.4-5        reshape2_1.4.3
## [61] R6_2.2.2           lubridate_1.7.4    knitr_1.20
## [64] dplyr_0.7.6        bindr_0.1.1        rprojroot_1.3-2
## [67] stringi_1.1.7      Rcpp_0.12.18       DEoptimR_1.0-8
## [70] tidyselect_0.2.4
```