Customer Churn Prediction

Madan K C

2023-04-27

Customer Churn Prediction

In this project, we will be analyzing a telecom churn dataset. The objective of the analysis is to develop a model to predict whether a customer will churn or not. The dataset consists of 7043 observations and 23 variables. The dataset contains a mix of categorical and continuous variables. I have taken the dataset from kaggle: https://github.com/madankc71/Customer-Churn-Prediction

I have determined generalized linear model with logistic regression and decision tree by implementing data partition and cross validation.

Loading the required package and reading the dataset

```
library(readxl)
telecom_data <- read_excel("data/telecom-churn-rate-dataset.xlsx")
#View(telecom_churn_rate_dataset)
```

Check the dimension of the dataset. Get the names of variables in the dataset

```
dim(telecom_data)
```

```
## [1] 7043   23
```

```
names(telecom_data)
```

```
##  [1] "customerID"       "gender"           "SeniorCitizen"    "Partner"
##  [5] "Dependents"       "tenure"           "PhoneService"     "MultipleLines"
##  [9] "InternetService"  "OnlineSecurity"   "OnlineBackup"     "DeviceProtection"
## [13] "TechSupport"      "StreamingTV"      "StreamingMovies"  "Contract"
## [17] "PaperlessBilling" "PaymentMethod"    "MonthlyCharges"   "TotalCharges"
## [21] "numAdminTickets"  "numTechTickets"   "Churn"
```

Get the structure of the dataset.

```
str(telecom_data)
```

```
## tibble [7,043 x 23] (S3: tbl_df/tbl/data.frame)
##  $ customerID      : chr [1:7043] "7590-VHVEG" "5575-GNVDE" "3668-QPYBK" "7795-CFOCW" ...
##  $ gender          : chr [1:7043] "Female" "Male" "Male" "Male" ...
##  $ SeniorCitizen   : num [1:7043] 0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ Partner         : chr [1:7043] "Yes" "No" "No" "No" ...
##  $ Dependents      : chr [1:7043] "No" "No" "No" "No" ...
##  $ tenure          : num [1:7043] 1 34 2 45 2 8 22 10 28 62 ...
##  $ PhoneService    : chr [1:7043] "No" "Yes" "Yes" "No" ...
##  $ MultipleLines   : chr [1:7043] "No phone service" "No" "No" "No phone service" ...
##  $ InternetService : chr [1:7043] "DSL" "DSL" "DSL" "DSL" ...
##  $ OnlineSecurity  : chr [1:7043] "No" "Yes" "Yes" "Yes" ...
##  $ OnlineBackup    : chr [1:7043] "Yes" "No" "Yes" "No" ...
##  $ DeviceProtection: chr [1:7043] "No" "Yes" "No" "Yes" ...
##  $ TechSupport     : chr [1:7043] "No" "No" "No" "Yes" ...
##  $ StreamingTV     : chr [1:7043] "No" "No" "No" "No" ...
##  $ StreamingMovies : chr [1:7043] "No" "No" "No" "No" ...
##  $ Contract        : chr [1:7043] "Month-to-month" "One year" "Month-to-month" "One year" ...
##  $ PaperlessBilling: chr [1:7043] "Yes" "No" "Yes" "No" ...
##  $ PaymentMethod   : chr [1:7043] "Electronic check" "Mailed check" "Mailed check" "Bank transfer (au
##  $ MonthlyCharges  : num [1:7043] 29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num [1:7043] 29.9 1889.5 108.2 1840.8 151.7 ...
##  $ numAdminTickets : num [1:7043] 0 0 0 0 0 0 0 0 0 0 ...
##  $ numTechTickets  : num [1:7043] 0 0 0 3 0 0 0 0 2 0 ...
##  $ Churn           : chr [1:7043] "No" "No" "Yes" "No" ...
```

There are 17 categorical (character) variables and 6 numeric variables in the dataset.

Find the number of unique values in each variable:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
sapply(telecom_data, n_distinct)
```

```
##       customerID           gender    SeniorCitizen          Partner
##             7043                2                2                2
##       Dependents           tenure     PhoneService    MultipleLines
##                2               73                2                3
##  InternetService   OnlineSecurity     OnlineBackup DeviceProtection
##                3                3                3                3
##      TechSupport      StreamingTV  StreamingMovies         Contract
##                3                3                3                3
## PaperlessBilling    PaymentMethod   MonthlyCharges     TotalCharges
##                2                4             1585             6531
##  numAdminTickets   numTechTickets            Churn
##                6               10                2
```

Check for missing values in the dataset:

```
colSums(is.na(telecom_data))
```

```
##       customerID           gender    SeniorCitizen          Partner
##                0                0                0                0
##       Dependents           tenure     PhoneService    MultipleLines
##                0                0                0                0
##  InternetService   OnlineSecurity     OnlineBackup DeviceProtection
##                0                0                0                0
##      TechSupport      StreamingTV  StreamingMovies         Contract
##                0                0                0                0
## PaperlessBilling    PaymentMethod   MonthlyCharges     TotalCharges
##                0                0                0               11
##  numAdminTickets   numTechTickets            Churn
##                0                0                0
```

The 'TotalCharges' variable shas 11 missing values

Remove the observations having missing values.

```
telecom_data <- na.omit(telecom_data)
```

Checking if there are any mising values again.

```
colSums(is.na(telecom_data))
```

```
##       customerID           gender    SeniorCitizen          Partner
##                0                0                0                0
##       Dependents           tenure     PhoneService    MultipleLines
##                0                0                0                0
##  InternetService   OnlineSecurity     OnlineBackup DeviceProtection
##                0                0                0                0
##      TechSupport      StreamingTV  StreamingMovies         Contract
##                0                0                0                0
## PaperlessBilling    PaymentMethod   MonthlyCharges     TotalCharges
##                0                0                0                0
##  numAdminTickets   numTechTickets            Churn
##                0                0                0
```

From this, we found that there are not any missing values.

For the logistic regression, it is appropriate to convert categorical variables to factor. Therefore, converting the categorical variables to factor.

```
telecom_data[, sapply(telecom_data, is.character)] <- lapply(telecom_data[, sapply(telecom_data, is.cha
```

Checking whether the categorical variables are converted to factor or not:

```
str(telecom_data)
```

```
## tibble [7,032 x 23] (S3: tbl_df/tbl/data.frame)
##  $ customerID      : Factor w/ 7032 levels "0002-ORFBO","0003-MKNFE",..: 5366 3954 2559 5525 6501 65
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
##  $ SeniorCitizen   : num [1:7032] 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          : num [1:7032] 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 [1:7032] 29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num [1:7032] 29.9 1889.5 108.2 1840.8 151.7 ...
##  $ numAdminTickets : num [1:7032] 0 0 0 0 0 0 0 0 0 0 ...
##  $ numTechTickets  : num [1:7032] 0 0 0 3 0 0 0 0 2 0 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:11] 489 754 937 1083 1341 3332 3827 4381 5219 6671 ...
##   ..- attr(*, "names")= chr [1:11] "489" "754" "937" "1083" ...
```

Grouping customers by gender and finding the number of customers for each gender:

```
library(magrittr)
library(dplyr)
customer_status <- telecom_data %>% group_by(gender) %>% summarise(num_customers = n())
customer_status
```

```
## # A tibble: 2 x 2
##   gender num_customers
##   <fct>          <int>
## 1 Female          3483
## 2 Male            3549
```

From the table, we found that there are 3483 female customers and 3549 male customers.

As 'Customer ID' variable is not related to the regression, so using variables except customerID variable. Removing customerID variable from the dataset:

```
telecom_data <- select(telecom_data, -customerID)
```

Logistic Regression:

Performing logistic regression with all the variables:

```r
logistic_reg1 <- glm(Churn ~ ., family = binomial, data = telecom_data)
summary(logistic_reg1)
```

```
##
## Call:
## glm(formula = Churn ~ ., family = binomial, data = telecom_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7730  -0.4598  -0.0797   0.2386   3.8509
##
## Coefficients: (7 not defined because of singularities)
##                                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           1.4285935  1.0245216   1.394 0.163197
## genderMale                           -0.0957337  0.0764895  -1.252 0.210718
## SeniorCitizen                         0.2698725  0.1045992   2.580 0.009878 **
## PartnerYes                           -0.0549046  0.0943263  -0.582 0.560519
## DependentsYes                        -0.0930284  0.1081090  -0.861 0.389511
## tenure                               -0.0683092  0.0072093  -9.475  < 2e-16 ***
## PhoneServiceYes                       0.2408302  0.8188660   0.294 0.768680
## MultipleLinesNo phone service                NA         NA      NA       NA
## MultipleLinesYes                      0.5188060  0.2209888   2.348 0.018892 *
## InternetServiceFiber optic            1.8314326  1.0126636   1.809 0.070524 .
## InternetServiceNo                    -1.9160048  1.0191497  -1.880 0.060108 .
## OnlineSecurityNo internet service            NA         NA      NA       NA
## OnlineSecurityYes                    -0.3581201  0.2254677  -1.588 0.112209
## OnlineBackupNo internet service              NA         NA      NA       NA
## OnlineBackupYes                      -0.2108974  0.2221889  -0.949 0.342529
## DeviceProtectionNo internet service          NA         NA      NA       NA
## DeviceProtectionYes                   0.0071828  0.2227764   0.032 0.974279
## TechSupportNo internet service               NA         NA      NA       NA
## TechSupportYes                       -0.0492909  0.2281475  -0.216 0.828950
## StreamingTVNo internet service               NA         NA      NA       NA
## StreamingTVYes                        0.3565925  0.4135539   0.862 0.388543
## StreamingMoviesNo internet service           NA         NA      NA       NA
## StreamingMoviesYes                    0.3587087  0.4142150   0.866 0.386492
## ContractOne year                     -0.8510954  0.1558515  -5.461 4.74e-08 ***
## ContractTwo year                     -2.4370384  0.3004879  -8.110 5.05e-16 ***
## PaperlessBillingYes                   0.3191673  0.0854006   3.737 0.000186 ***
## PaymentMethodCredit card (automatic) -0.2092096  0.1431671  -1.461 0.143934
## PaymentMethodElectronic check         0.1326731  0.1182345   1.122 0.261812
## PaymentMethodMailed check            -0.2372223  0.1337243  -1.774 0.076069 .
## MonthlyCharges                       -0.0355461  0.0402637  -0.883 0.377327
## TotalCharges                         -0.0002083  0.0000880  -2.367 0.017947 *
## numAdminTickets                      -0.0514773  0.0300145  -1.715 0.086330 .
## numTechTickets                        1.4598345  0.0532417  27.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8143.4  on 7031  degrees of freedom
## Residual deviance: 4257.2  on 7006  degrees of freedom
```

```
## AIC: 4309.2
##
## Number of Fisher Scoring iterations: 7
```

There are several values with multicollinearity and insignificantly large p-values.

Considering only the variables having significant p-value. Performing logistic regression with significant variables only:

```
logistic_reg <- glm(Churn ~ SeniorCitizen + tenure + MultipleLines + Contract + PaperlessBilling + Total
summary(logistic_reg)
```

```
##
## Call:
## glm(formula = Churn ~ SeniorCitizen + tenure + MultipleLines +
##     Contract + PaperlessBilling + TotalCharges + numTechTickets,
##     family = binomial, data = telecom_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9517  -0.5189  -0.0862   0.2388   4.0840
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -2.756e-01  7.308e-02  -3.771 0.000162 ***
## SeniorCitizen                5.774e-01  9.761e-02   5.916 3.30e-09 ***
## tenure                      -9.959e-02  6.643e-03 -14.993  < 2e-16 ***
## MultipleLinesNo phone service  3.416e-01  1.300e-01   2.628 0.008578 **
## MultipleLinesYes             6.188e-01  8.869e-02   6.977 3.01e-12 ***
## ContractOne year            -1.308e+00  1.484e-01  -8.811  < 2e-16 ***
## ContractTwo year            -2.992e+00  2.877e-01 -10.399  < 2e-16 ***
## PaperlessBillingYes          6.563e-01  7.910e-02   8.298  < 2e-16 ***
## TotalCharges                 1.961e-04  7.336e-05   2.674 0.007505 **
## numTechTickets               1.412e+00  5.122e-02  27.576  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8143.4  on 7031  degrees of freedom
## Residual deviance: 4579.8  on 7022  degrees of freedom
## AIC: 4599.8
##
## Number of Fisher Scoring iterations: 7
```

Thus, by determining appropriate generalized linear models, I have met the second objective (Determine and apply the appropriate generalized linear model for a specific data context).

Objective 3: Conduct model selection for a set of candidate models

Data Partition and Modelling

In this chunk of code, the necessary packages are loaded, and the telecom data is partitioned into a training and testing set. The leave column is created as a factor with two levels, "Churn" and "Not Churn" and the "Churn" column is removed. The glm function from caret is used to fit a logistic regression model to the training set. The model's performance is then evaluated using confusion matrices for both the training and testing sets.

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
data_partition <- telecom_data
#telecom_data$Churn = as.factor(telecom_data$Churn)
data_partition$leave = ifelse(data_partition$Churn == "Yes","Churn","Not Churn")
data_partition$leave = as.factor(data_partition$leave)
data_partition = data_partition %>% dplyr::select(-Churn) #removing the column with numbers, otherwise

set.seed(1)
test.indices = createDataPartition(data_partition$leave, p = 0.2, list = FALSE) #classic 80/20 train-te
test_partition = data_partition[test.indices,]
train_partition = data_partition[-test.indices,]
```

The code fits a generalized linear model (GLM) using the train function from the caret package to predict customer churn based on seven predictor variables. The trained model is then used to generate churn predictions for both the training and test partitions.

```
model_train = train(leave ~ SeniorCitizen + tenure + MultipleLines + Contract + PaperlessBilling + Tota

predTrain = predict(model_train, train_partition)
predTest = predict(model_train, test_partition)
```

For Training data, the Confusion Matrix:

```
confusionMatrix(predTrain, train_partition$leave, positive = "Churn")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Churn Not Churn
##   Churn       997       367
##   Not Churn   498      3763
##
##                Accuracy : 0.8462
##                  95% CI : (0.8365, 0.8556)
##     No Information Rate : 0.7342
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.5946
##
##  Mcnemar's Test P-Value : 9.864e-06
##
##             Sensitivity : 0.6669
##             Specificity : 0.9111
##          Pos Pred Value : 0.7309
##          Neg Pred Value : 0.8831
##              Prevalence : 0.2658
##          Detection Rate : 0.1772
##    Detection Prevalence : 0.2425
##       Balanced Accuracy : 0.7890
##
##        'Positive' Class : Churn
##
```

We got 84.62% accuracy here.

For Testing data, the Confusion Matrix:

```
confusionMatrix(predTest, test_partition$leave, positive = "Churn")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Churn Not Churn
##   Churn       253        92
##   Not Churn   121       941
##
##                Accuracy : 0.8486
##                  95% CI : (0.8288, 0.867)
##     No Information Rate : 0.7342
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6023
##
##  Mcnemar's Test P-Value : 0.05504
##
##             Sensitivity : 0.6765
##             Specificity : 0.9109
##          Pos Pred Value : 0.7333
##          Neg Pred Value : 0.8861
##              Prevalence : 0.2658
##          Detection Rate : 0.1798
##    Detection Prevalence : 0.2452
##       Balanced Accuracy : 0.7937
##
##        'Positive' Class : Churn
##
```

For the testing data, the accuracy is little more than for the training data (84.86%) which is a good prediction.

Cross-validation

In this chunk of code, a 15-fold cross-validation technique is applied to the logistic regression model previously built. The performance of the model is then printed. The model's performance is also evaluated using a confusion matrix on the testing set.

```
train_control <- trainControl(method="cv", number=15) #15-fold cross validation
model_cv <- caret::train(leave ~ SeniorCitizen + tenure + MultipleLines + Contract + PaperlessBilling +
print(model_cv)
```

```
## Generalized Linear Model
##
## 5625 samples
##    7 predictor
##    2 classes: 'Churn', 'Not Churn'
##
## No pre-processing
## Resampling: Cross-Validated (15 fold)
## Summary of sample sizes: 5251, 5250, 5250, 5250, 5250, 5250, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8444451  0.5910879
```

We got the accuracy of 84.45% which is less than that of data partition we did before.

Now, finding the accuracy for the test data:

```
predTest.cv <- predict(model_cv, test_partition)
cmTest.cv = confusionMatrix(predTest.cv, test_partition$leave)
cmTest.cv
```

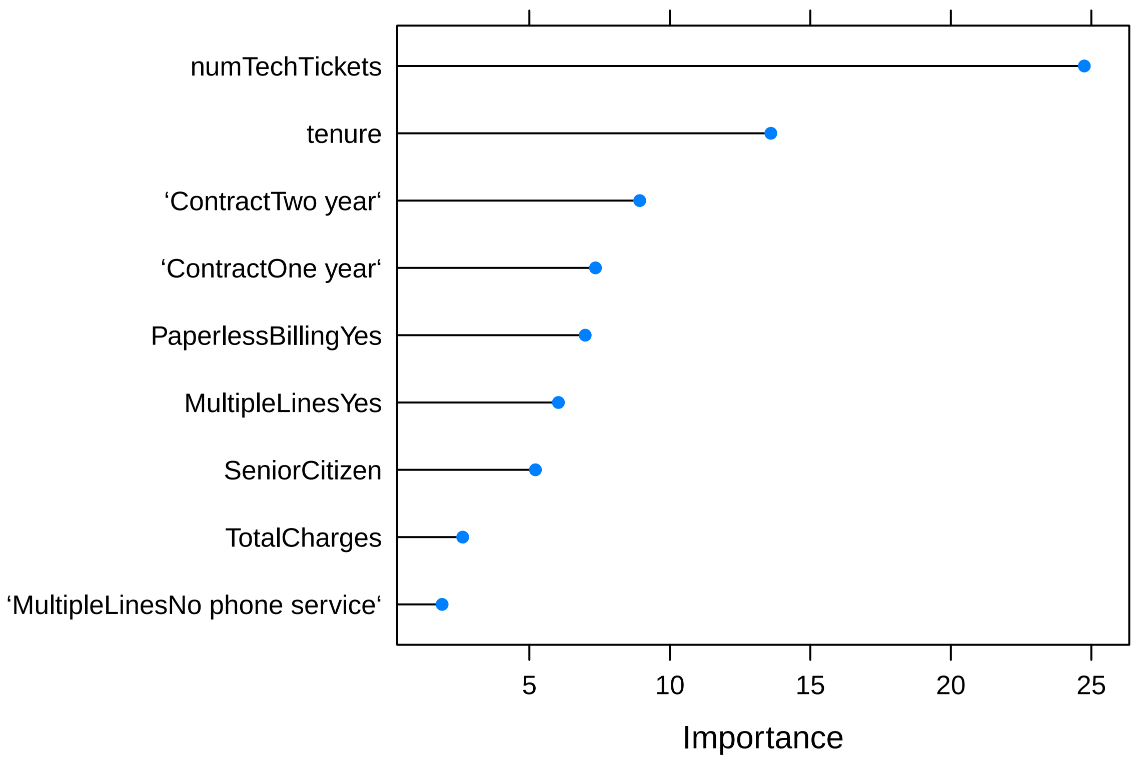
```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Churn Not Churn
##   Churn       253        92
##   Not Churn   121       941
##
##                Accuracy : 0.8486
##                  95% CI : (0.8288, 0.867)
##     No Information Rate : 0.7342
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6023
##
##  Mcnemar's Test P-Value : 0.05504
##
##             Sensitivity : 0.6765
##             Specificity : 0.9109
##          Pos Pred Value : 0.7333
##          Neg Pred Value : 0.8861
##              Prevalence : 0.2658
```

```
##          Detection Rate : 0.1798
##    Detection Prevalence : 0.2452
##       Balanced Accuracy : 0.7937
##
##        'Positive' Class : Churn
##
```

The accuracy is 84.86% which is equal to that of normal data partition.

Now finding the important variables for the model:

```
importance <- varImp(model_train, scale=FALSE)
plot(importance)
```



Decision Tree

In this chunk of code, the party package is loaded, and a decision tree is built using the ctree function. The model's performance is evaluated using confusion matrices for both the training and testing sets.

```
library(party)
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

##
## Attaching package: 'party'

## The following object is masked from 'package:dplyr':
##
##     where
```

```
tree <- ctree(leave~., data = train_partition)
```

```
treePredTrain <- predict(tree, train_partition, type = "response")

confusionMatrix(treePredTrain,train_partition$leave)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Churn Not Churn
##   Churn      1078       337
##   Not Churn   417      3793
##
##                Accuracy : 0.866
##                  95% CI : (0.8568, 0.8748)
##     No Information Rate : 0.7342
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6506
##
##  Mcnemar's Test P-Value : 0.004015
##
##             Sensitivity : 0.7211
##             Specificity : 0.9184
##          Pos Pred Value : 0.7618
##          Neg Pred Value : 0.9010
##              Prevalence : 0.2658
```

```
##          Detection Rate : 0.1916
##    Detection Prevalence : 0.2516
##       Balanced Accuracy : 0.8197
##
##        'Positive' Class : Churn
##
```

We got the 86.6% accuracy using decision tree which is greater than others above.

Finding accuracy on the test data:

```r
treePredTest <- predict(tree, test_partition, type = "response")

confusionMatrix(treePredTest,test_partition$leave)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Churn Not Churn
##   Churn       259       100
##   Not Churn   115       933
##
##                Accuracy : 0.8472
##                  95% CI : (0.8273, 0.8656)
##     No Information Rate : 0.7342
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6034
##
##  Mcnemar's Test P-Value : 0.3397
##
##             Sensitivity : 0.6925
##             Specificity : 0.9032
##          Pos Pred Value : 0.7214
##          Neg Pred Value : 0.8903
##              Prevalence : 0.2658
##          Detection Rate : 0.1841
##    Detection Prevalence : 0.2552
##       Balanced Accuracy : 0.7979
##
##        'Positive' Class : Churn
##
```

The accuracy we got is the largest till now: 84.72%.

Cross Validation: Decision Tree

In this chunk of code, a 10-fold cross-validation technique is applied to the decision tree model previously built. The performance of the model is then printed. The model's performance is also evaluated using a confusion matrix on the testing set.

Overall, the code performs data partitioning, logistic regression, cross-validation, and decision tree modeling on the telecom data set and evaluates the model's performance on the training and testing sets.

```r
train_control_tree <- trainControl(method="cv", number=15)
model_tree <- caret::train(leave~., data=train_partition, trControl=train_control_tree, method="ctree")

print(model_tree)
```

```
## Conditional Inference Tree
##
## 5625 samples
##   21 predictor
##    2 classes: 'Churn', 'Not Churn'
##
## No pre-processing
## Resampling: Cross-Validated (15 fold)
## Summary of sample sizes: 5250, 5250, 5250, 5250, 5249, 5250, ...
## Resampling results across tuning parameters:
##
##   mincriterion  Accuracy   Kappa
##   0.01          0.8435433  0.5934086
##   0.50          0.8463896  0.5952389
##   0.99          0.8515490  0.6181477
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mincriterion = 0.99.
```

The final model had a mincriterion value of 0.99 and an accuracy of 0.8488913.

```r
predTest_tree <- predict(model_tree, test_partition)
tree_cv = confusionMatrix(predTest_tree, test_partition$leave)
tree_cv
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Churn Not Churn
##   Churn       260       109
##   Not Churn   114       924
##
##                Accuracy : 0.8415
##                  95% CI : (0.8214, 0.8602)
##     No Information Rate : 0.7342
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.5922
##
##  Mcnemar's Test P-Value : 0.7888
##
##             Sensitivity : 0.6952
##             Specificity : 0.8945
##          Pos Pred Value : 0.7046
##          Neg Pred Value : 0.8902
##              Prevalence : 0.2658
##          Detection Rate : 0.1848
```

```
##    Detection Prevalence : 0.2623
##       Balanced Accuracy : 0.7948
##
##        'Positive' Class : Churn
##
```

However, the prediction on the test data decreases to 84.15%.

Therefore, we are selecting decision tree with data partition into 80/20 (train/test) among logistic regression with data partition and cross validation and decision tree with data partition and cross validation. The selected model has 86.6% accuracy in train and 86.72% accuracy in the test data.