**GROUP PROJECT REPORT**

**ON**

**CHURN DATA PREDICTION IN**

**TELECOM INDUSTRY**

Logo, company name

Description automatically generated

**Instructor Group-3**

**Dambar Uprety, Ph.D. Pratheek Sreerangam**

**Business Analytics Pavan Chaitanya.B**

**MIS 64036-002 Pravalika Girneni**

**Contributions**

|  |  |
| --- | --- |
| **NAMES** | **CONTRIBUTION** |
| Pratheek Sreerangam  psreeran@kent.edu | Model Building, Model Performance, Predictions and Results, Data Cleaning, Data Exploration, Documentation and Presentation |
| Pavan Chaitanya.B  pbommade@kent.edu | Model Building, Model Performance, Predictions and Results, Data Cleaning, Data Exploration, Documentation and Presentation |
| Pravalika Girneni  pgirneni@kent.edu | Model Building, Model Performance, Predictions and Results, Data Cleaning, Data Exploration, Documentation and Presentation |

**Index**

* Project Goal
* Overview of the data
  + Data Exploration Analysis
* Model Strategy
  + Techniques Used (Decision Tree)
* Model Performance
  + Performance Metrics
  + AUC Model
* Insights and Conclusion

**Project Goal**

Churn is a challenge for telecom companies considering that it is more difficult to attract new customers than it is to keep onto their present clients. Customer churn modelling has been quite popular recently since there are signs that a significant portion of a company's revenue comes from repeat consumers. Businesses are also very interested in figuring out which consumers are likely to leave, and they commonly use data mining tools to do this. Using the available data, we were able to identify clients who were most likely to abandon this project and provide them with sufficient inducements to do so.

This project's objective is to employ a predictive model to analyse data and spot trends in order to foretell when a regular client would transfer service providers. Our investigation may be carried out using a variety of prediction models, including regression. Here, we'll build our model using a decision tree classifier.

**Overview of Data**

ABC wireless company has provided the following data from which we can infer:

Demographics

* + State
  + Account length
  + Area code
  + International plan
  + Voice-mail plan

Calling Behaviour

* + Number of messages
  + Total day minutes
  + Total day calls
  + Total day charge
  + Total evening minutes
  + Total evening calls
  + Total evening charges
  + Total night minutes
  + Total night calls
  + Total night charges
  + Total International minutes
  + Total International calls
  + Total International charges
  + Number of calls to customer service

**Exploratory Analysis**

**Part 1: Churn Data**

Data # Loading the required Libraries that are required for the Project.

**library**(readr)

## Warning: package 'readr' was built under R version 4.2.2

**library**(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.2.2

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──

## ✔ ggplot2 3.3.6 ✔ dplyr 1.0.10

## ✔ tibble 3.1.8 ✔ stringr 1.4.1

## ✔ tidyr 1.2.1 ✔ forcats 0.5.2

## ✔ purrr 0.3.5

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──

## ✖ dplyr::filter() masks stats::filter()

## ✖ dplyr::lag() masks stats::lag()

**library**(caret)

## Loading required package: lattice

##

## Attaching package: 'caret'

##

## The following object is masked from 'package:purrr':

##

## lift

**library**(gmodels)

**library**(rpart)

## Warning: package 'rpart' was built under R version 4.2.2

**library**(pROC)

## Warning: package 'pROC' was built under R version 4.2.2

## Type 'citation("pROC")' for a citation.

##

## Attaching package: 'pROC'

##

## The following object is masked from 'package:gmodels':

##

## ci

##

## The following objects are masked from 'package:stats':

##

## cov, smooth, var

**library**(rattle)

## Warning: package 'rattle' was built under R version 4.2.2

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.

## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.

## Type 'rattle()' to shake, rattle, and roll your data.

# **Importing the Churn Dataset that is given to us.**

Given\_Churn\_Datafile= read.csv("C:/Users/girne/Downloads/Churn\_Train.csv")

# **Examining the details regarding the data file.**

*# Head Part of the Data file*

head(Given\_Churn\_Datafile)

## state account\_length area\_code international\_plan voice\_mail\_plan

## 1 NV 125 area\_code\_510 no no

## 2 HI 108 area\_code\_415 no no

## 3 DC 82 area\_code\_415 no no

## 4 HI NA area\_code\_408 no yes

## 5 OH 83 area\_code\_415 no no

## 6 MO 89 area\_code\_415 no no

## number\_vmail\_messages total\_day\_minutes total\_day\_calls total\_day\_charge

## 1 0 2013.4 99 28.66

## 2 0 291.6 99 49.57

## 3 0 300.3 109 51.05

## 4 30 110.3 71 18.75

## 5 0 337.4 120 57.36

## 6 0 178.7 81 30.38

## total\_eve\_minutes total\_eve\_calls total\_eve\_charge total\_night\_minutes

## 1 1107.6 107 14.93 243.3

## 2 221.1 93 18.79 229.2

## 3 181.0 100 15.39 270.1

## 4 182.4 108 15.50 183.8

## 5 227.4 116 19.33 153.9

## 6 NA 74 19.86 131.9

## total\_night\_calls total\_night\_charge total\_intl\_minutes total\_intl\_calls

## 1 92 10.95 10.9 7

## 2 110 10.31 14.0 9

## 3 73 12.15 11.7 4

## 4 88 8.27 11.0 8

## 5 114 6.93 15.8 7

## 6 120 5.94 9.1 4

## total\_intl\_charge number\_customer\_service\_calls churn

## 1 2.94 0 no

## 2 3.78 2 yes

## 3 3.16 0 yes

## 4 2.97 2 no

## 5 4.27 0 yes

## 6 2.46 1 no

*#Summary of the Data present in the data file.*

summary(Given\_Churn\_Datafile)

## state account\_length area\_code international\_plan

## Length:3333 Min. :-209.00 Length:3333 Length:3333

## Class :character 1st Qu.: 72.00 Class :character Class :character

## Mode :character Median : 100.00 Mode :character Mode :character

## Mean : 97.32

## 3rd Qu.: 127.00

## Max. : 243.00

## NA's :501

## voice\_mail\_plan number\_vmail\_messages total\_day\_minutes total\_day\_calls

## Length:3333 Min. :-10.000 Min. : 0.0 Min. : 0.0

## Class :character 1st Qu.: 0.000 1st Qu.: 149.3 1st Qu.: 87.0

## Mode :character Median : 0.000 Median : 190.5 Median :101.0

## Mean : 7.333 Mean : 418.9 Mean :100.3

## 3rd Qu.: 16.000 3rd Qu.: 237.8 3rd Qu.:114.0

## Max. : 51.000 Max. :2185.1 Max. :165.0

## NA's :200 NA's :200 NA's :200

## total\_day\_charge total\_eve\_minutes total\_eve\_calls total\_eve\_charge

## Min. : 0.00 Min. : 0.0 Min. : 0.0 Min. : 0.00

## 1st Qu.:24.45 1st Qu.: 170.5 1st Qu.: 87.0 1st Qu.:14.14

## Median :30.65 Median : 209.9 Median :100.0 Median :17.09

## Mean :30.63 Mean : 324.3 Mean :100.1 Mean :17.08

## 3rd Qu.:36.84 3rd Qu.: 257.6 3rd Qu.:114.0 3rd Qu.:20.00

## Max. :59.64 Max. :1244.2 Max. :170.0 Max. :30.91

## NA's :200 NA's :301 NA's :200 NA's :200

## total\_night\_minutes total\_night\_calls total\_night\_charge total\_intl\_minutes

## Min. : 23.2 Min. : 33.0 Min. : 1.040 Min. : 0.00

## 1st Qu.:167.3 1st Qu.: 87.0 1st Qu.: 7.530 1st Qu.: 8.50

## Median :201.4 Median :100.0 Median : 9.060 Median :10.30

## Mean :201.2 Mean :100.1 Mean : 9.054 Mean :10.23

## 3rd Qu.:235.3 3rd Qu.:113.0 3rd Qu.:10.590 3rd Qu.:12.10

## Max. :395.0 Max. :175.0 Max. :17.770 Max. :20.00

## NA's :200 NA's :200 NA's :200

## total\_intl\_calls total\_intl\_charge number\_customer\_service\_calls

## Min. : 0.00 Min. :0.000 Min. :0.000

## 1st Qu.: 3.00 1st Qu.:2.300 1st Qu.:1.000

## Median : 4.00 Median :2.780 Median :1.000

## Mean : 4.47 Mean :2.762 Mean :1.561

## 3rd Qu.: 6.00 3rd Qu.:3.270 3rd Qu.:2.000

## Max. :20.00 Max. :5.400 Max. :9.000

## NA's :301 NA's :200 NA's :200

## churn

## Length:3333

## Class :character

## Mode :character

##

##

##

##

*#Data Types of Data Columns in the Data file*

str(Given\_Churn\_Datafile)

## 'data.frame': 3333 obs. of 20 variables:

## $ state : chr "NV" "HI" "DC" "HI" ...

## $ account\_length : int 125 108 82 NA 83 89 135 28 86 65 ...

## $ area\_code : chr "area\_code\_510" "area\_code\_415" "area\_code\_415" "area\_code\_408" ...

## $ international\_plan : chr "no" "no" "no" "no" ...

## $ voice\_mail\_plan : chr "no" "no" "no" "yes" ...

## $ number\_vmail\_messages : int 0 0 0 30 0 0 0 0 0 0 ...

## $ total\_day\_minutes : num 2013 292 300 110 337 ...

## $ total\_day\_calls : int 99 99 109 71 120 81 81 87 115 137 ...

## $ total\_day\_charge : num 28.7 49.6 51 18.8 57.4 ...

## $ total\_eve\_minutes : num 1108 221 181 182 227 ...

## $ total\_eve\_calls : int 107 93 100 108 116 74 114 92 112 83 ...

## $ total\_eve\_charge : num 14.9 18.8 15.4 15.5 19.3 ...

## $ total\_night\_minutes : num 243 229 270 184 154 ...

## $ total\_night\_calls : int 92 110 73 88 114 120 82 112 95 111 ...

## $ total\_night\_charge : num 10.95 10.31 12.15 8.27 6.93 ...

## $ total\_intl\_minutes : num 10.9 14 11.7 11 15.8 9.1 10.3 10.1 9.8 12.7 ...

## $ total\_intl\_calls : int 7 9 4 8 7 4 6 3 7 6 ...

## $ total\_intl\_charge : num 2.94 3.78 3.16 2.97 4.27 2.46 2.78 2.73 2.65 3.43 ...

## $ number\_customer\_service\_calls: int 0 2 0 2 0 1 1 3 2 4 ...

## $ churn : chr "no" "yes" "yes" "no" ...

*#Glimpse of the Data Given to us*

glimpse(Given\_Churn\_Datafile)

## Rows: 3,333

## Columns: 20

## $ state <chr> "NV", "HI", "DC", "HI", "OH", "MO", "NC"…

## $ account\_length <int> 125, 108, 82, NA, 83, 89, 135, 28, 86, 6…

## $ area\_code <chr> "area\_code\_510", "area\_code\_415", "area\_…

## $ international\_plan <chr> "no", "no", "no", "no", "no", "no", "no"…

## $ voice\_mail\_plan <chr> "no", "no", "no", "yes", "no", "no", "no…

## $ number\_vmail\_messages <int> 0, 0, 0, 30, 0, 0, 0, 0, 0, 0, 0, NA, 32…

## $ total\_day\_minutes <dbl> 2013.4, 291.6, 300.3, 110.3, 337.4, 178.…

## $ total\_day\_calls <int> 99, 99, 109, 71, 120, 81, 81, 87, 115, 1…

## $ total\_day\_charge <dbl> 28.66, 49.57, 51.05, 18.75, 57.36, 30.38…

## $ total\_eve\_minutes <dbl> 1107.6, 221.1, 181.0, 182.4, 227.4, NA, …

## $ total\_eve\_calls <int> 107, 93, 100, 108, 116, 74, 114, 92, 112…

## $ total\_eve\_charge <dbl> 14.93, 18.79, 15.39, 15.50, 19.33, 19.86…

## $ total\_night\_minutes <dbl> 243.3, 229.2, 270.1, 183.8, 153.9, 131.9…

## $ total\_night\_calls <int> 92, 110, 73, 88, 114, 120, 82, 112, 95, …

## $ total\_night\_charge <dbl> 10.95, 10.31, 12.15, 8.27, 6.93, 5.94, 9…

## $ total\_intl\_minutes <dbl> 10.9, 14.0, 11.7, 11.0, 15.8, 9.1, 10.3,…

## $ total\_intl\_calls <int> 7, 9, 4, 8, 7, 4, 6, 3, 7, 6, 7, NA, 4, …

## $ total\_intl\_charge <dbl> 2.94, 3.78, 3.16, 2.97, 4.27, 2.46, 2.78…

## $ number\_customer\_service\_calls <int> 0, 2, 0, 2, 0, 1, 1, 3, 2, 4, 1, NA, 3, …

## $ churn <chr> "no", "yes", "yes", "no", "yes", "no", "…

# **Data Type Conversion.**

*# Converting the Char type data to factors for our convience*

Given\_Churn\_Datafile = Given\_Churn\_Datafile %>% mutate\_if(is.character, as.factor)

# **Checking where the data conversion is successful or not.**

str(Given\_Churn\_Datafile)

## 'data.frame': 3333 obs. of 20 variables:

## $ state : Factor w/ 51 levels "AK","AL","AR",..: 34 12 8 12 36 25 28 39 13 16 ...

## $ account\_length : int 125 108 82 NA 83 89 135 28 86 65 ...

## $ area\_code : Factor w/ 3 levels "area\_code\_408",..: 3 2 2 1 2 2 2 2 1 2 ...

## $ international\_plan : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

## $ voice\_mail\_plan : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...

## $ number\_vmail\_messages : int 0 0 0 30 0 0 0 0 0 0 ...

## $ total\_day\_minutes : num 2013 292 300 110 337 ...

## $ total\_day\_calls : int 99 99 109 71 120 81 81 87 115 137 ...

## $ total\_day\_charge : num 28.7 49.6 51 18.8 57.4 ...

## $ total\_eve\_minutes : num 1108 221 181 182 227 ...

## $ total\_eve\_calls : int 107 93 100 108 116 74 114 92 112 83 ...

## $ total\_eve\_charge : num 14.9 18.8 15.4 15.5 19.3 ...

## $ total\_night\_minutes : num 243 229 270 184 154 ...

## $ total\_night\_calls : int 92 110 73 88 114 120 82 112 95 111 ...

## $ total\_night\_charge : num 10.95 10.31 12.15 8.27 6.93 ...

## $ total\_intl\_minutes : num 10.9 14 11.7 11 15.8 9.1 10.3 10.1 9.8 12.7 ...

## $ total\_intl\_calls : int 7 9 4 8 7 4 6 3 7 6 ...

## $ total\_intl\_charge : num 2.94 3.78 3.16 2.97 4.27 2.46 2.78 2.73 2.65 3.43 ...

## $ number\_customer\_service\_calls: int 0 2 0 2 0 1 1 3 2 4 ...

## $ churn : Factor w/ 2 levels "no","yes": 1 2 2 1 2 1 1 1 1 2 ...

# **Checking for the NA values if they are present in the dataset.**

colSums(is.na(Given\_Churn\_Datafile))

## state account\_length

## 0 501

## area\_code international\_plan

## 0 0

## voice\_mail\_plan number\_vmail\_messages

## 0 200

## total\_day\_minutes total\_day\_calls

## 200 200

## total\_day\_charge total\_eve\_minutes

## 200 301

## total\_eve\_calls total\_eve\_charge

## 200 200

## total\_night\_minutes total\_night\_calls

## 200 0

## total\_night\_charge total\_intl\_minutes

## 200 200

## total\_intl\_calls total\_intl\_charge

## 301 200

## number\_customer\_service\_calls churn

## 200 0

# **Checking for the Negative Values if they are present in dataset by columns wise.**

sapply(Given\_Churn\_Datafile %>% select\_if(is.numeric), **function**(x) {

sum(x < 0, na.rm = TRUE)

})

## account\_length number\_vmail\_messages

## 51 201

## total\_day\_minutes total\_day\_calls

## 0 0

## total\_day\_charge total\_eve\_minutes

## 0 0

## total\_eve\_calls total\_eve\_charge

## 0 0

## total\_night\_minutes total\_night\_calls

## 0 0

## total\_night\_charge total\_intl\_minutes

## 0 0

## total\_intl\_calls total\_intl\_charge

## 0 0

## number\_customer\_service\_calls

## 0

Given\_Churn\_Datafile =

Given\_Churn\_Datafile %>% mutate\_if(is.numeric, **function**(x) {

ifelse(x < 0, abs(x), x)

})

*# We see that account\_length and number\_vmail\_messages have some Negative values and we cannot remove them because they are connected to the final Churn Variable.*

# **To deal with NA Values which are present in the data and removing them from the data set.**

*# We are following the MedianImpute as a Method to dela with the NA Values in the Dataset*

NA\_Dealing\_Model= preProcess(Given\_Churn\_Datafile %>% select\_if(is.numeric),method = “medianImpute”)

Predict\_Data = predict(NA\_Dealing\_Model, Given\_Churn\_Datafile %>% select\_if(is.numeric))

Given\_Churn\_Datafile = Given\_Churn\_Datafile %>% select(setdiff(names(Given\_Churn\_Datafile), names(Predict\_Data))) %>% cbind(Predict\_Data)

*# Viewing the Datafile with no NA Values*

view(Given\_Churn\_Datafile)

*# Checking Finally wether there are any NA Values Present in the each Column of the dataset.*

colSums(is.na(Given\_Churn\_Datafile))

## state area\_code

## 0 0

## international\_plan voice\_mail\_plan

## 0 0

## churn account\_length

## 0 0

## number\_vmail\_messages total\_day\_minutes

## 0 0

## total\_day\_calls total\_day\_charge

## 0 0

## total\_eve\_minutes total\_eve\_calls

## 0 0

## total\_eve\_charge total\_night\_minutes

## 0 0

## total\_night\_calls total\_night\_charge

## 0 0

## total\_intl\_minutes total\_intl\_calls

## 0 0

## total\_intl\_charge number\_customer\_service\_calls

## 0 0

# **Visualization of the Data present in the Dataset**

*# Numeric Values Distribution Plot*

Given\_Churn\_Datafile %>% select\_if(is.numeric) %>% mutate\_all(scale) %>% gather(“features”,”values”) %>% na.omit() %>%

ggplot(aes(x = features, y = values)) +

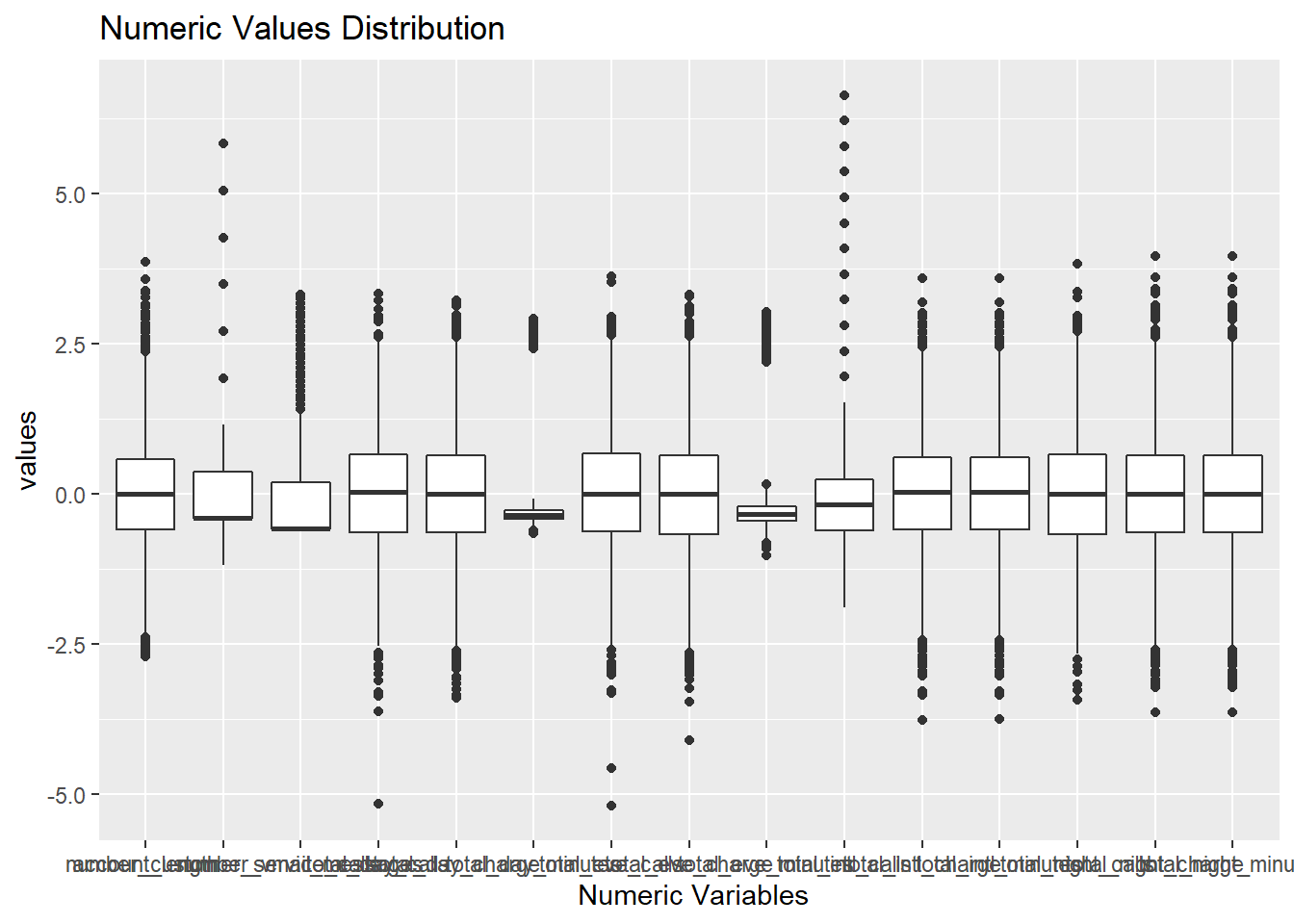
geom\_boxplot(show.legend = FALSE) +

labs(x = “ Numeric Variables”) +

ggtitle(label = “Numeric Values Distribution”)

## Warning: attributes are not identical across measure variables;

## they will be dropped



*# Churn Variable Visualization*

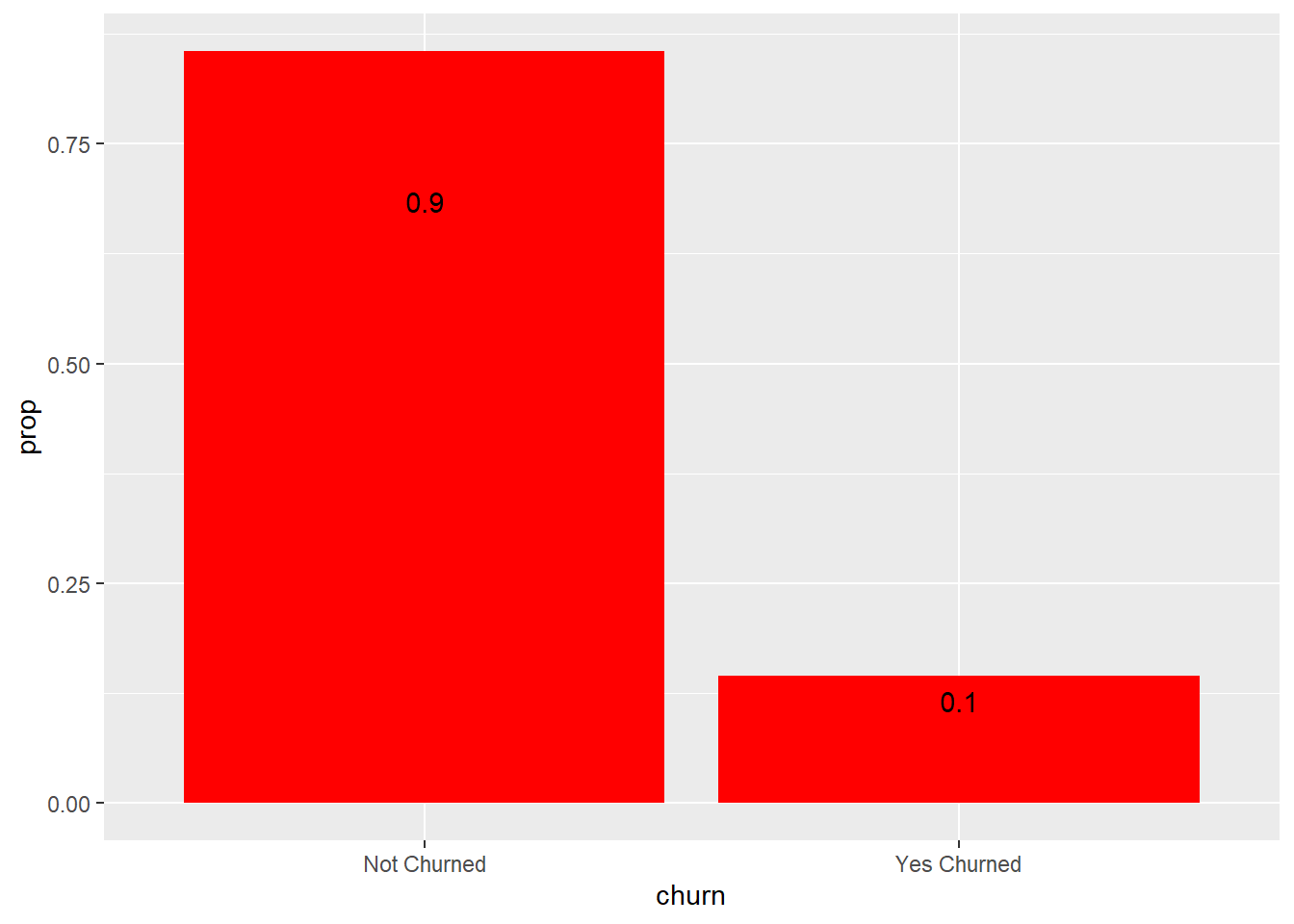
ggplot(Given\_Churn\_Datafile, aes(x=churn, y=..prop..,group = 2)) +

geom\_bar(fill=”Red”) +

geom\_text(aes(label=round(..prop..,1)),stat = “count”,

position = position\_stack(vjust=0.8)) +

scale\_x\_discrete(labels = c(“Not Churned”,”Yes Churned”))



*# From the Plot we can see that 90 % hasn’t churned but 10 % churned.*

# **Adding the State and Churn Variables to the Updated Churn Dataset for our calculations.**

Str(Given\_Churn\_Datafile) *# Without Updation*

## ‘data.frame’: 3333 obs. Of 20 variables:

## $ state : Factor w/ 51 levels “AK”,”AL”,”AR”,..: 34 12 8 12 36 25 28 39 13 16 …

## $ area\_code : Factor w/ 3 levels “area\_code\_408”,..: 3 2 2 1 2 2 2 2 1 2 …

## $ international\_plan : Factor w/ 2 levels “no”,”yes”: 1 1 1 1 1 1 1 1 1 1 …

## $ voice\_mail\_plan : Factor w/ 2 levels “no”,”yes”: 1 1 1 2 1 1 1 1 1 1 …

## $ churn : Factor w/ 2 levels “no”,”yes”: 1 2 2 1 2 1 1 1 1 2 …

## $ account\_length : num 125 108 82 101 83 89 135 28 86 65 …

## $ number\_vmail\_messages : num 0 0 0 30 0 0 0 0 0 0 …

## $ total\_day\_minutes : num 2013 292 300 110 337 …

## $ total\_day\_calls : num 99 99 109 71 120 81 81 87 115 137 …

## $ total\_day\_charge : num 28.7 49.6 51 18.8 57.4 …

## $ total\_eve\_minutes : num 1108 221 181 182 227 …

## $ total\_eve\_calls : num 107 93 100 108 116 74 114 92 112 83 …

## $ total\_eve\_charge : num 14.9 18.8 15.4 15.5 19.3 …

## $ total\_night\_minutes : num 243 229 270 184 154 …

## $ total\_night\_calls : int 92 110 73 88 114 120 82 112 95 111 …

## $ total\_night\_charge : num 10.95 10.31 12.15 8.27 6.93 …

## $ total\_intl\_minutes : num 10.9 14 11.7 11 15.8 9.1 10.3 10.1 9.8 12.7 …

## $ total\_intl\_calls : num 7 9 4 8 7 4 6 3 7 6 …

## $ total\_intl\_charge : num 2.94 3.78 3.16 2.97 4.27 2.46 2.78 2.73 2.65 3.43 …

## $ number\_customer\_service\_calls: num 0 2 0 2 0 1 1 3 2 4 …

Given\_Churn\_Datafile = Given\_Churn\_Datafile %>% select(-state, -churn) %>%

fastDummies::dummy\_cols(., remove\_selected\_columns = TRUE) %>% mutate(state = Given\_Churn\_Datafile$state, churn = Given\_Churn\_Datafile$churn)

str(Given\_Churn\_Datafile) *# With Updation*

## ‘data.frame’: 3333 obs. Of 24 variables:

## $ account\_length : num 125 108 82 101 83 89 135 28 86 65 …

## $ number\_vmail\_messages : num 0 0 0 30 0 0 0 0 0 0 …

## $ total\_day\_minutes : num 2013 292 300 110 337 …

## $ total\_day\_calls : num 99 99 109 71 120 81 81 87 115 137 …

## $ total\_day\_charge : num 28.7 49.6 51 18.8 57.4 …

## $ total\_eve\_minutes : num 1108 221 181 182 227 …

## $ total\_eve\_calls : num 107 93 100 108 116 74 114 92 112 83 …

## $ total\_eve\_charge : num 14.9 18.8 15.4 15.5 19.3 …

## $ total\_night\_minutes : num 243 229 270 184 154 …

## $ total\_night\_calls : int 92 110 73 88 114 120 82 112 95 111 …

## $ total\_night\_charge : num 10.95 10.31 12.15 8.27 6.93 …

## $ total\_intl\_minutes : num 10.9 14 11.7 11 15.8 9.1 10.3 10.1 9.8 12.7 …

## $ total\_intl\_calls : num 7 9 4 8 7 4 6 3 7 6 …

## $ total\_intl\_charge : num 2.94 3.78 3.16 2.97 4.27 2.46 2.78 2.73 2.65 3.43 …

## $ number\_customer\_service\_calls: num 0 2 0 2 0 1 1 3 2 4 …

## $ area\_code\_area\_code\_408 : int 0 0 0 1 0 0 0 0 1 0 …

## $ area\_code\_area\_code\_415 : int 0 1 1 0 1 1 1 1 0 1 …

## $ area\_code\_area\_code\_510 : int 1 0 0 0 0 0 0 0 0 0 …

## $ international\_plan\_no : int 1 1 1 1 1 1 1 1 1 1 …

## $ international\_plan\_yes : int 0 0 0 0 0 0 0 0 0 0 …

## $ voice\_mail\_plan\_no : int 1 1 1 0 1 1 1 1 1 1 …

## $ voice\_mail\_plan\_yes : int 0 0 0 1 0 0 0 0 0 0 …

## $ state : Factor w/ 51 levels “AK”,”AL”,”AR”,..: 34 12 8 12 36 25 28 39 13 16 …

## $ churn : Factor w/ 2 levels “no”,”yes”: 1 2 2 1 2 1 1 1 1 2 …

# **Model Strategy**

# **What Technique: we are following the Decision tree as our Model.**

# **Why: We believe that to illustrate the influence of numerous variables and their significance in forecasting the result of the target variable, so we will go with Decision Tree approach.**

# **Preprocessing of Data:**

*# Splitting the dataset into training set(75%) and validation set(25%).*

set.seed(5454)

Data\_partition<- createDataPartition(Given\_Churn\_Datafile$churn, p=0.75, list=FALSE)

Req\_Churn\_Data\_train = Given\_Churn\_Datafile[Data\_partition,]

Req\_Churn\_Data\_test = Given\_Churn\_Datafile[-Data\_partition,]

# **Scaling the Preprocessed Data**

PreProcess\_Scale <- preProcess(Req\_Churn\_Data\_train %>% select\_if(is.numeric), method = c("center", "scale"))

Req\_Churn\_Data\_train\_norm <- predict(PreProcess\_Scale, Req\_Churn\_Data\_train %>% select\_if(is.numeric))

Req\_Churn\_Data\_test\_norm <- predict(PreProcess\_Scale, Req\_Churn\_Data\_test %>% select\_if(is.numeric))

Req\_Churn\_Data\_train\_norm$churn <- Req\_Churn\_Data\_train$churn

Req\_Churn\_Data\_test\_norm$churn <- Req\_Churn\_Data\_test$churn

# **Model Construction**

*# Using Rplot*

DecisionTree\_Model <- rpart(churn ~ ., data = Req\_Churn\_Data\_train\_norm, method = "class")

summary(DecisionTree\_Model)

## Call:

## rpart(formula = churn ~ ., data = Req\_Churn\_Data\_train\_norm,

## method = "class")

## n= 2501

##

## CP nsplit rel error xerror xstd

## 1 0.08402204 0 1.0000000 1.0000000 0.04852815

## 2 0.05922865 2 0.8319559 0.8016529 0.04417526

## 3 0.05234160 4 0.7134986 0.6997245 0.04161548

## 4 0.01652893 8 0.4793388 0.5206612 0.03641341

## 5 0.01239669 10 0.4462810 0.4931129 0.03551356

## 6 0.01101928 12 0.4214876 0.4986226 0.03569602

## 7 0.01000000 14 0.3994490 0.4903581 0.03542184

##

## Variable importance

## total\_day\_charge number\_customer\_service\_calls

## 21 11

## total\_eve\_charge international\_plan\_no

## 8 7

## international\_plan\_yes total\_intl\_charge

## 7 7

## total\_intl\_minutes total\_day\_minutes

## 7 7

## total\_intl\_calls total\_eve\_minutes

## 6 5

## number\_vmail\_messages voice\_mail\_plan\_no

## 4 4

## voice\_mail\_plan\_yes total\_night\_calls

## 4 1

##

## Node number 1: 2501 observations, complexity param=0.08402204

## predicted class=no expected loss=0.1451419 P(node) =1

## class counts: 2138 363

## probabilities: 0.855 0.145

## left son=2 (2308 obs) right son=3 (193 obs)

## Primary splits:

## number\_customer\_service\_calls < 1.523388 to the left, improve=61.47075, (0 missing)

## total\_day\_charge < 1.621606 to the left, improve=59.79091, (0 missing)

## international\_plan\_no < -1.318779 to the right, improve=49.47426, (0 missing)

## international\_plan\_yes < 1.318779 to the left, improve=49.47426, (0 missing)

## total\_day\_minutes < -0.2493636 to the left, improve=18.28591, (0 missing)

##

## Node number 2: 2308 observations, complexity param=0.05922865

## predicted class=no expected loss=0.1130849 P(node) =0.9228309

## class counts: 2047 261

## probabilities: 0.887 0.113

## left son=4 (2078 obs) right son=5 (230 obs)

## Primary splits:

## total\_day\_charge < 1.247929 to the left, improve=61.79721, (0 missing)

## international\_plan\_no < -1.318779 to the right, improve=49.35911, (0 missing)

## international\_plan\_yes < 1.318779 to the left, improve=49.35911, (0 missing)

## total\_day\_minutes < -0.2879089 to the left, improve=25.10998, (0 missing)

## total\_eve\_charge < 0.8901874 to the left, improve= 7.79800, (0 missing)

##

## Node number 3: 193 observations, complexity param=0.08402204

## predicted class=yes expected loss=0.4715026 P(node) =0.07716913

## class counts: 91 102

## probabilities: 0.472 0.528

## left son=6 (118 obs) right son=7 (75 obs)

## Primary splits:

## total\_day\_charge < -0.3672269 to the right, improve=35.086420, (0 missing)

## total\_day\_minutes < -0.3915621 to the right, improve=31.762260, (0 missing)

## total\_eve\_charge < 0.2318583 to the right, improve= 8.112675, (0 missing)

## total\_eve\_minutes < -0.3205428 to the right, improve= 7.129213, (0 missing)

## total\_night\_calls < -1.075241 to the right, improve= 4.779043, (0 missing)

## Surrogate splits:

## total\_day\_minutes < -0.3915621 to the right, agree=0.969, adj=0.920, (0 split)

## total\_night\_calls < -1.075241 to the right, agree=0.637, adj=0.067, (0 split)

## total\_night\_minutes < -2.275635 to the right, agree=0.627, adj=0.040, (0 split)

## total\_night\_charge < -2.276326 to the right, agree=0.627, adj=0.040, (0 split)

## number\_customer\_service\_calls < 3.082464 to the left, agree=0.627, adj=0.040, (0 split)

##

## Node number 4: 2078 observations, complexity param=0.0523416

## predicted class=no expected loss=0.07459095 P(node) =0.8308677

## class counts: 1923 155

## probabilities: 0.925 0.075

## left son=8 (1883 obs) right son=9 (195 obs)

## Primary splits:

## international\_plan\_no < -1.318779 to the right, improve=42.746610, (0 missing)

## international\_plan\_yes < 1.318779 to the left, improve=42.746610, (0 missing)

## total\_day\_charge < 0.8109463 to the left, improve= 4.897006, (0 missing)

## total\_intl\_minutes < 1.083145 to the left, improve= 4.231993, (0 missing)

## total\_intl\_charge < 1.081839 to the left, improve= 4.231993, (0 missing)

## Surrogate splits:

## international\_plan\_yes < 1.318779 to the left, agree=1.000, adj=1.00, (0 split)

## total\_day\_charge < 1.233363 to the left, agree=0.907, adj=0.01, (0 split)

##

## Node number 5: 230 observations, complexity param=0.05922865

## predicted class=no expected loss=0.4608696 P(node) =0.09196321

## class counts: 124 106

## probabilities: 0.539 0.461

## left son=10 (117 obs) right son=11 (113 obs)

## Primary splits:

## total\_eve\_charge < 0.0717242 to the left, improve=23.37878, (0 missing)

## voice\_mail\_plan\_yes < 0.5001899 to the right, improve=21.78033, (0 missing)

## voice\_mail\_plan\_no < -0.5001899 to the left, improve=21.78033, (0 missing)

## number\_vmail\_messages < 0.1466111 to the right, improve=21.11552, (0 missing)

## total\_eve\_minutes < -0.3578247 to the left, improve=19.57100, (0 missing)

## Surrogate splits:

## total\_eve\_minutes < -0.3471728 to the left, agree=0.926, adj=0.850, (0 split)

## total\_night\_calls < -0.4545841 to the left, agree=0.565, adj=0.115, (0 split)

## total\_intl\_minutes < 0.7323531 to the left, agree=0.561, adj=0.106, (0 split)

## total\_intl\_charge < 0.7331038 to the left, agree=0.561, adj=0.106, (0 split)

## total\_day\_calls < 0.1489096 to the right, agree=0.548, adj=0.080, (0 split)

##

## Node number 6: 118 observations, complexity param=0.01652893

## predicted class=no expected loss=0.2881356 P(node) =0.04718113

## class counts: 84 34

## probabilities: 0.712 0.288

## left son=12 (96 obs) right son=13 (22 obs)

## Primary splits:

## total\_eve\_charge < -0.9139902 to the right, improve=6.558295, (0 missing)

## total\_eve\_minutes < -0.5097817 to the right, improve=6.086780, (0 missing)

## total\_day\_charge < 2.01545 to the left, improve=4.818620, (0 missing)

## total\_night\_calls < 0.3988196 to the left, improve=3.859411, (0 missing)

## total\_day\_calls < -0.1573803 to the left, improve=1.707479, (0 missing)

## Surrogate splits:

## total\_eve\_minutes < -0.5097817 to the right, agree=0.966, adj=0.818, (0 split)

## total\_night\_calls < -1.902784 to the right, agree=0.831, adj=0.091, (0 split)

##

## Node number 7: 75 observations

## predicted class=yes expected loss=0.09333333 P(node) =0.029988

## class counts: 7 68

## probabilities: 0.093 0.907

##

## Node number 8: 1883 observations, complexity param=0.01239669

## predicted class=no expected loss=0.04195433 P(node) =0.7528988

## class counts: 1804 79

## probabilities: 0.958 0.042

## left son=16 (1714 obs) right son=17 (169 obs)

## Primary splits:

## total\_day\_charge < 0.8507229 to the left, improve=4.1702330, (0 missing)

## total\_eve\_charge < 1.348052 to the left, improve=2.7665920, (0 missing)

## total\_day\_minutes < -0.3505868 to the left, improve=1.5914910, (0 missing)

## total\_eve\_minutes < -0.3321934 to the left, improve=1.1171860, (0 missing)

## total\_night\_minutes < -0.7620966 to the left, improve=0.7805677, (0 missing)

##

## Node number 9: 195 observations, complexity param=0.0523416

## predicted class=no expected loss=0.3897436 P(node) =0.07796881

## class counts: 119 76

## probabilities: 0.610 0.390

## left son=18 (157 obs) right son=19 (38 obs)

## Primary splits:

## total\_intl\_calls < -0.8236005 to the right, improve=35.153880, (0 missing)

## total\_intl\_minutes < 1.064683 to the left, improve=27.454100, (0 missing)

## total\_intl\_charge < 1.061325 to the left, improve=27.454100, (0 missing)

## total\_night\_minutes < 1.419998 to the right, improve= 2.082097, (0 missing)

## total\_night\_charge < 1.419451 to the right, improve= 2.082097, (0 missing)

##

## Node number 10: 117 observations, complexity param=0.01652893

## predicted class=no expected loss=0.2393162 P(node) =0.04678129

## class counts: 89 28

## probabilities: 0.761 0.239

## left son=20 (109 obs) right son=21 (8 obs)

## Primary splits:

## total\_day\_charge < 2.503975 to the left, improve=6.940034, (0 missing)

## total\_day\_minutes < -0.1931379 to the left, improve=5.792412, (0 missing)

## total\_night\_minutes < 1.070244 to the left, improve=5.233092, (0 missing)

## total\_night\_charge < 1.068673 to the left, improve=5.233092, (0 missing)

## number\_vmail\_messages < 0.0320373 to the right, improve=3.616295, (0 missing)

## Surrogate splits:

## account\_length < 2.534459 to the left, agree=0.949, adj=0.25, (0 split)

##

## Node number 11: 113 observations, complexity param=0.0523416

## predicted class=yes expected loss=0.3097345 P(node) =0.04518193

## class counts: 35 78

## probabilities: 0.310 0.690

## left son=22 (25 obs) right son=23 (88 obs)

## Primary splits:

## voice\_mail\_plan\_no < -0.5001899 to the left, improve=20.879490, (0 missing)

## voice\_mail\_plan\_yes < 0.5001899 to the right, improve=20.879490, (0 missing)

## number\_vmail\_messages < 0.1848024 to the right, improve=18.101190, (0 missing)

## total\_day\_minutes < -0.2166002 to the left, improve= 5.371216, (0 missing)

## total\_day\_charge < 1.621606 to the left, improve= 4.406838, (0 missing)

## Surrogate splits:

## voice\_mail\_plan\_yes < 0.5001899 to the right, agree=1.000, adj=1.00, (0 split)

## number\_vmail\_messages < 0.1848024 to the right, agree=0.982, adj=0.92, (0 split)

## total\_eve\_minutes < 3.001706 to the right, agree=0.788, adj=0.04, (0 split)

## total\_eve\_calls < 1.902658 to the right, agree=0.788, adj=0.04, (0 split)

##

## Node number 12: 96 observations, complexity param=0.01101928

## predicted class=no expected loss=0.2083333 P(node) =0.03838465

## class counts: 76 20

## probabilities: 0.792 0.208

## left son=24 (82 obs) right son=25 (14 obs)

## Primary splits:

## total\_day\_charge < 1.599756 to the left, improve=6.189315, (0 missing)

## total\_night\_calls < 0.3988196 to the left, improve=3.760417, (0 missing)

## total\_day\_minutes < -0.2185274 to the left, improve=2.483568, (0 missing)

## international\_plan\_yes < 1.318779 to the left, improve=1.190476, (0 missing)

## international\_plan\_no < -1.318779 to the right, improve=1.190476, (0 missing)

## Surrogate splits:

## total\_day\_minutes < -0.2185274 to the left, agree=0.885, adj=0.214, (0 split)

##

## Node number 13: 22 observations, complexity param=0.01101928

## predicted class=yes expected loss=0.3636364 P(node) =0.008796481

## class counts: 8 14

## probabilities: 0.364 0.636

## left son=26 (12 obs) right son=27 (10 obs)

## Primary splits:

## total\_day\_minutes < -0.3324035 to the right, improve=4.848485, (0 missing)

## total\_day\_charge < 0.3050545 to the right, improve=4.848485, (0 missing)

## total\_intl\_calls < -0.3986753 to the right, improve=2.715152, (0 missing)

## total\_eve\_calls < 0.1973581 to the right, improve=2.548485, (0 missing)

## number\_customer\_service\_calls < 2.302926 to the left, improve=1.000866, (0 missing)

## Surrogate splits:

## total\_day\_charge < 0.3050545 to the right, agree=1.000, adj=1.0, (0 split)

## total\_eve\_calls < -0.694645 to the right, agree=0.682, adj=0.3, (0 split)

## total\_night\_calls < 0.7091483 to the left, agree=0.682, adj=0.3, (0 split)

## total\_intl\_calls < -0.8236005 to the right, agree=0.682, adj=0.3, (0 split)

## number\_customer\_service\_calls < 2.302926 to the left, agree=0.682, adj=0.3, (0 split)

##

## Node number 16: 1714 observations

## predicted class=no expected loss=0.03150525 P(node) =0.6853259

## class counts: 1660 54

## probabilities: 0.968 0.032

##

## Node number 17: 169 observations, complexity param=0.01239669

## predicted class=no expected loss=0.147929 P(node) =0.06757297

## class counts: 144 25

## probabilities: 0.852 0.148

## left son=34 (148 obs) right son=35 (21 obs)

## Primary splits:

## total\_eve\_charge < 1.336191 to the left, improve=15.383470, (0 missing)

## total\_eve\_minutes < -0.1381279 to the left, improve= 8.862374, (0 missing)

## total\_day\_calls < 1.323021 to the left, improve= 2.963844, (0 missing)

## number\_vmail\_messages < -0.006153971 to the right, improve= 2.488166, (0 missing)

## voice\_mail\_plan\_yes < 0.5001899 to the right, improve= 2.244367, (0 missing)

## Surrogate splits:

## total\_eve\_minutes < -0.1381279 to the left, agree=0.923, adj=0.381, (0 split)

##

## Node number 18: 157 observations, complexity param=0.0523416

## predicted class=no expected loss=0.2420382 P(node) =0.06277489

## class counts: 119 38

## probabilities: 0.758 0.242

## left son=36 (129 obs) right son=37 (28 obs)

## Primary splits:

## total\_intl\_minutes < 1.064683 to the left, improve=39.155480, (0 missing)

## total\_intl\_charge < 1.061325 to the left, improve=39.155480, (0 missing)

## account\_length < 0.02805502 to the right, improve= 1.923262, (0 missing)

## total\_night\_minutes < 0.2830391 to the right, improve= 1.894086, (0 missing)

## total\_night\_charge < 0.2822885 to the right, improve= 1.894086, (0 missing)

## Surrogate splits:

## total\_intl\_charge < 1.061325 to the left, agree=1.000, adj=1.000, (0 split)

## number\_vmail\_messages < 2.552661 to the left, agree=0.834, adj=0.071, (0 split)

## total\_day\_minutes < -0.5673619 to the right, agree=0.834, adj=0.071, (0 split)

## total\_day\_charge < -2.419366 to the right, agree=0.834, adj=0.071, (0 split)

##

## Node number 19: 38 observations

## predicted class=yes expected loss=0 P(node) =0.01519392

## class counts: 0 38

## probabilities: 0.000 1.000

##

## Node number 20: 109 observations

## predicted class=no expected loss=0.1926606 P(node) =0.04358257

## class counts: 88 21

## probabilities: 0.807 0.193

##

## Node number 21: 8 observations

## predicted class=yes expected loss=0.125 P(node) =0.003198721

## class counts: 1 7

## probabilities: 0.125 0.875

##

## Node number 22: 25 observations

## predicted class=no expected loss=0.12 P(node) =0.009996002

## class counts: 22 3

## probabilities: 0.880 0.120

##

## Node number 23: 88 observations

## predicted class=yes expected loss=0.1477273 P(node) =0.03518593

## class counts: 13 75

## probabilities: 0.148 0.852

##

## Node number 24: 82 observations

## predicted class=no expected loss=0.1341463 P(node) =0.03278689

## class counts: 71 11

## probabilities: 0.866 0.134

##

## Node number 25: 14 observations

## predicted class=yes expected loss=0.3571429 P(node) =0.005597761

## class counts: 5 9

## probabilities: 0.357 0.643

##

## Node number 26: 12 observations

## predicted class=no expected loss=0.3333333 P(node) =0.004798081

## class counts: 8 4

## probabilities: 0.667 0.333

##

## Node number 27: 10 observations

## predicted class=yes expected loss=0 P(node) =0.003998401

## class counts: 0 10

## probabilities: 0.000 1.000

##

## Node number 34: 148 observations

## predicted class=no expected loss=0.06756757 P(node) =0.05917633

## class counts: 138 10

## probabilities: 0.932 0.068

##

## Node number 35: 21 observations

## predicted class=yes expected loss=0.2857143 P(node) =0.008396641

## class counts: 6 15

## probabilities: 0.286 0.714

##

## Node number 36: 129 observations

## predicted class=no expected loss=0.07751938 P(node) =0.05157937

## class counts: 119 10

## probabilities: 0.922 0.078

##

## Node number 37: 28 observations

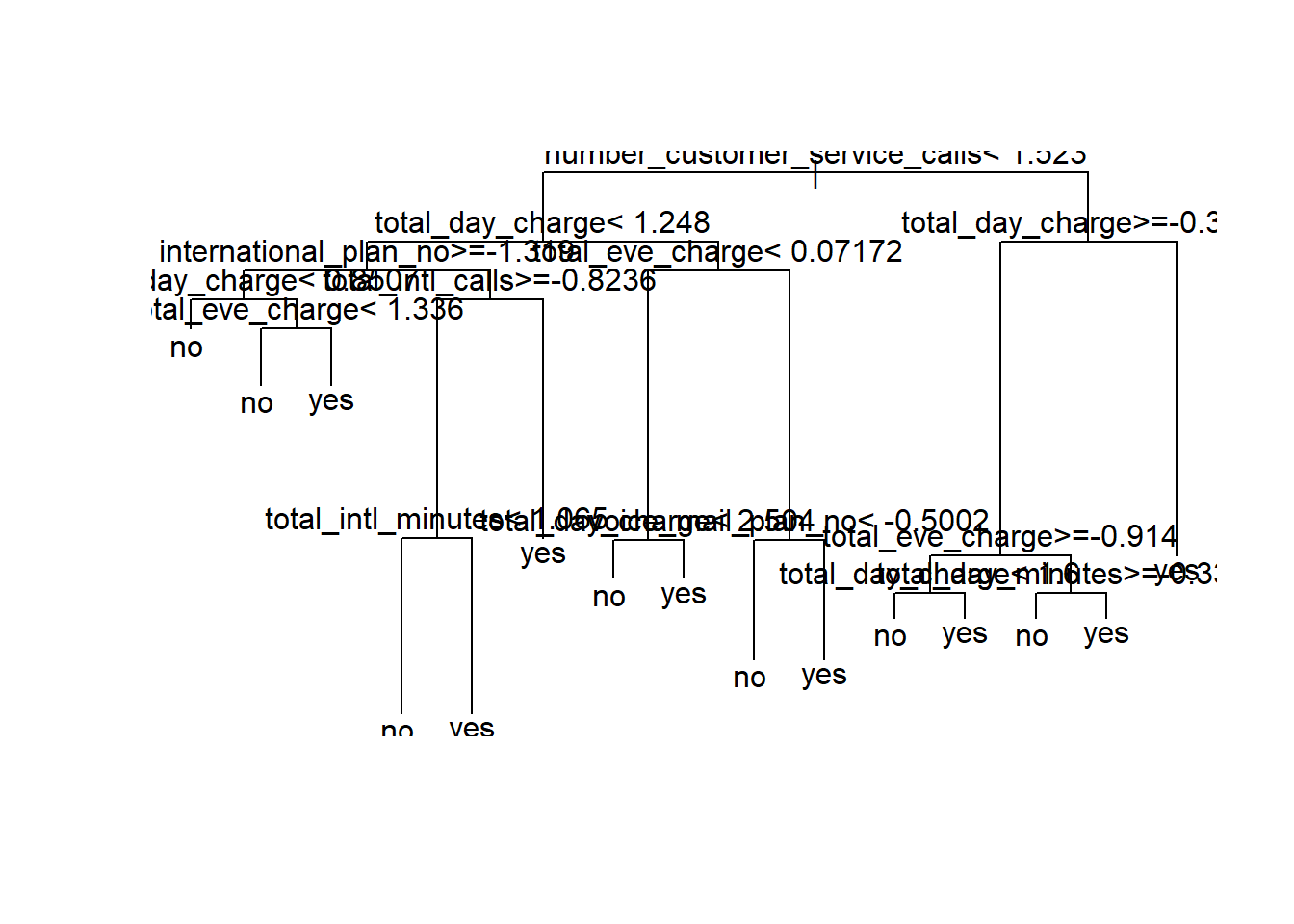
## predicted class=yes expected loss=0 P(node) =0.01119552

## class counts: 0 28

## probabilities: 0.000 1.000

plot(DecisionTree\_Model)

text(DecisionTree\_Model)



print(DecisionTree\_Model)

## n= 2501

##

## node), split, n, loss, yval, (yprob)

## \* denotes terminal node

##

## 1) root 2501 363 no (0.85485806 0.14514194)

## 2) number\_customer\_service\_calls< 1.523388 2308 261 no (0.88691508 0.11308492)

## 4) total\_day\_charge< 1.247929 2078 155 no (0.92540905 0.07459095)

## 8) international\_plan\_no>=-1.318779 1883 79 no (0.95804567 0.04195433)

## 16) total\_day\_charge< 0.8507229 1714 54 no (0.96849475 0.03150525) \*

## 17) total\_day\_charge>=0.8507229 169 25 no (0.85207101 0.14792899)

## 34) total\_eve\_charge< 1.336191 148 10 no (0.93243243 0.06756757) \*

## 35) total\_eve\_charge>=1.336191 21 6 yes (0.28571429 0.71428571) \*

## 9) international\_plan\_no< -1.318779 195 76 no (0.61025641 0.38974359)

## 18) total\_intl\_calls>=-0.8236005 157 38 no (0.75796178 0.24203822)

## 36) total\_intl\_minutes< 1.064683 129 10 no (0.92248062 0.07751938) \*

## 37) total\_intl\_minutes>=1.064683 28 0 yes (0.00000000 1.00000000) \*

## 19) total\_intl\_calls< -0.8236005 38 0 yes (0.00000000 1.00000000) \*

## 5) total\_day\_charge>=1.247929 230 106 no (0.53913043 0.46086957)

## 10) total\_eve\_charge< 0.0717242 117 28 no (0.76068376 0.23931624)

## 20) total\_day\_charge< 2.503975 109 21 no (0.80733945 0.19266055) \*

## 21) total\_day\_charge>=2.503975 8 1 yes (0.12500000 0.87500000) \*

## 11) total\_eve\_charge>=0.0717242 113 35 yes (0.30973451 0.69026549)

## 22) voice\_mail\_plan\_no< -0.5001899 25 3 no (0.88000000 0.12000000) \*

## 23) voice\_mail\_plan\_no>=-0.5001899 88 13 yes (0.14772727 0.85227273) \*

## 3) number\_customer\_service\_calls>=1.523388 193 91 yes (0.47150259 0.52849741)

## 6) total\_day\_charge>=-0.3672269 118 34 no (0.71186441 0.28813559)

## 12) total\_eve\_charge>=-0.9139902 96 20 no (0.79166667 0.20833333)

## 24) total\_day\_charge< 1.599756 82 11 no (0.86585366 0.13414634) \*

## 25) total\_day\_charge>=1.599756 14 5 yes (0.35714286 0.64285714) \*

## 13) total\_eve\_charge< -0.9139902 22 8 yes (0.36363636 0.63636364)

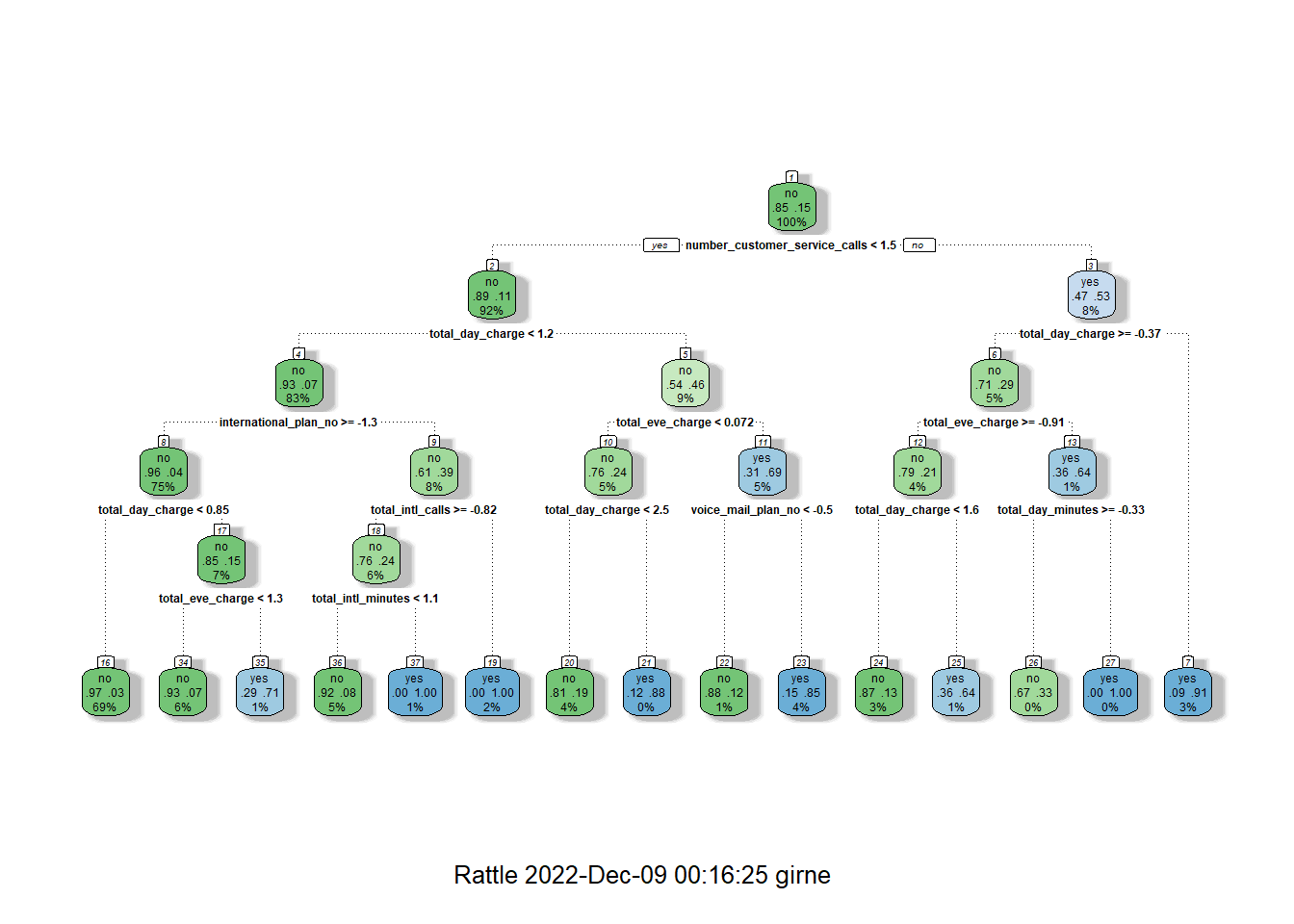
## 26) total\_day\_minutes>=-0.3324035 12 4 no (0.66666667 0.33333333) \*

## 27) total\_day\_minutes< -0.3324035 10 0 yes (0.00000000 1.00000000) \*

## 7) total\_day\_charge< -0.3672269 75 7 yes (0.09333333 0.90666667) \*

*# Using fancyRpartPlot*

fancyRpartPlot(DecisionTree\_Model)



# **Model Performance**

# **Model Building is done and we can interpret the results.**

*# Predicting values using based on DecisionTree\_Model.*

pred\_labels <- predict(object = DecisionTree\_Model,Req\_Churn\_Data\_test\_norm, type = "class")

pred\_probs <- predict(object = DecisionTree\_Model,Req\_Churn\_Data\_test\_norm)

*# Performance Metrics*

*# Confusion matrix for the DecisionTree\_Model.*

CrossTable(x=Req\_Churn\_Data\_test\_norm$churn, y = pred\_labels, prop.chisq = FALSE)

##

##

## Cell Contents

## |-------------------------|

## | N |

## | N / Row Total |

## | N / Col Total |

## | N / Table Total |

## |-------------------------|

##

##

## Total Observations in Table: 832

##

##

## | pred\_labels

## Req\_Churn\_Data\_test\_norm$churn | no | yes | Row Total |

## -------------------------------|-----------|-----------|-----------|

## no | 700 | 12 | 712 |

## | 0.983 | 0.017 | 0.856 |

## | 0.932 | 0.148 | |

## | 0.841 | 0.014 | |

## -------------------------------|-----------|-----------|-----------|

## yes | 51 | 69 | 120 |

## | 0.425 | 0.575 | 0.144 |

## | 0.068 | 0.852 | |

## | 0.061 | 0.083 | |

## -------------------------------|-----------|-----------|-----------|

## Column Total | 751 | 81 | 832 |

## | 0.903 | 0.097 | |

## -------------------------------|-----------|-----------|-----------|

##

##

confusionMatrix(pred\_labels,Req\_Churn\_Data\_test\_norm$churn)

## Confusion Matrix and Statistics

##

## Reference

## Prediction no yes

## no 700 51

## yes 12 69

##

## Accuracy : 0.9243

## 95% CI : (0.9042, 0.9413)

## No Information Rate : 0.8558

## P-Value [Acc > NIR] : 8.126e-10

##

## Kappa : 0.6453

##

## Mcnemar's Test P-Value : 1.688e-06

##

## Sensitivity : 0.9831

## Specificity : 0.5750

## Pos Pred Value : 0.9321

## Neg Pred Value : 0.8519

## Prevalence : 0.8558

## Detection Rate : 0.8413

## Detection Prevalence : 0.9026

## Balanced Accuracy : 0.7791

##

## 'Positive' Class : no

##

*# From the confusion Matrix we can say that*

*# Accuracy ~ 0.93*

*# Sensitivity ~ 0.95*

*# Specificity ~0.6*

# **AUC of the Model**

roc(Req\_Churn\_Data\_test$churn, pred\_probs[,2])

## Setting levels: control = no, case = yes

## Setting direction: controls < cases

##

## Call:

## roc.default(response = Req\_Churn\_Data\_test$churn, predictor = pred\_probs[, 2])

##

## Data: pred\_probs[, 2] in 712 controls (Req\_Churn\_Data\_test$churn no) < 120 cases (Req\_Churn\_Data\_test$churn yes).

## Area under the curve: 0.8702

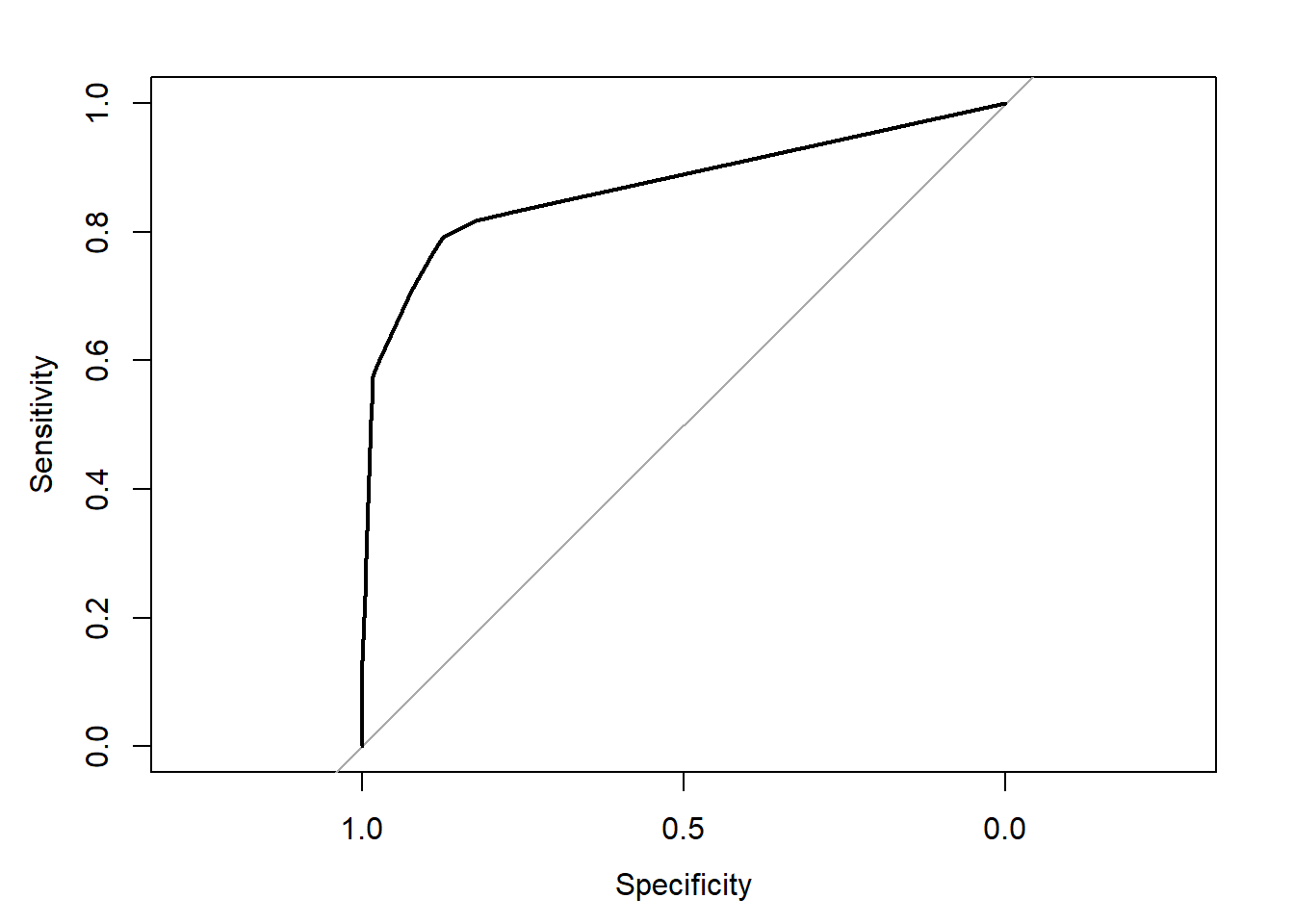
*# As AUC is greater than 0.8 we can say that the model is good.*

*# Plotting the AUC of the Model*

plot.roc(roc(Req\_Churn\_Data\_test$churn, pred\_probs[,2]))

## Setting levels: control = no, case = yes

## Setting direction: controls < cases



**Conclusion:**

We used the decision Tree classifier as our model and found out the AUC and Accuracy. As AUC is above 0.8, we can say that our model is Excellent.

**Insights:**

 Part 2: Predicting for Customers\_To\_Predict

*# We need to use load() to read the RData file*

load("C:/Users/girne/Downloads/Customers\_To\_Predict (1).RData")

Customers\_To\_Predict\_data <- Customers\_To\_Predict

Customers\_To\_Predict <- Customers\_To\_Predict %>% select(-state) %>% fastDummies::dummy\_cols(., remove\_selected\_columns = TRUE)

Customers\_To\_Predict <- as.data.frame(scale(Customers\_To\_Predict))

predict\_labels <- predict(object = DecisionTree\_Model, Customers\_To\_Predict, type = "class")

*# Adding the New Predicting column to the Customer\_To\_Predict Datafile.*

Customers\_To\_Predict <- Customers\_To\_Predict\_data %>% mutate(Churn\_Probability = predict\_labels)

*# Viewing the Updated Data File*

View(Customers\_To\_Predict)

*#Head Part of the Updated Data file*

head(Customers\_To\_Predict)

## # A tibble: 6 × 20

## state accoun…¹ area\_…² inter…³ voice…⁴ numbe…⁵ total…⁶ total…⁷ total…⁸ total…⁹

## <chr> <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>

## 1 UT 93 area\_c… no no 0 174. 127 29.6 177.

## 2 SD 39 area\_c… no no 0 179 88 30.4 148.

## 3 KY 124 area\_c… no no 0 157. 74 26.7 196.

## 4 MS 162 area\_c… yes no 0 172. 138 29.3 166.

## 5 AK 112 area\_c… no yes 31 143. 92 24.3 234.

## 6 TX 109 area\_c… yes no 0 160. 136 27.1 151

## # … with 10 more variables: total\_eve\_calls <dbl>, total\_eve\_charge <dbl>,

## # total\_night\_minutes <dbl>, total\_night\_calls <dbl>,

## # total\_night\_charge <dbl>, total\_intl\_minutes <dbl>, total\_intl\_calls <dbl>,

## # total\_intl\_charge <dbl>, number\_customer\_service\_calls <dbl>,

## # Churn\_Probability <fct>, and abbreviated variable names ¹​account\_length,

## # ²​area\_code, ³​international\_plan, ⁴​voice\_mail\_plan, ⁵​number\_vmail\_messages,

## # ⁶​total\_day\_minutes, ⁷​total\_day\_calls, ⁸​total\_day\_charge, …

*#Printing only the Churn\_Probability Column*

print(Customers\_To\_Predict$Churn\_Probability)

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

## no no no no no yes no no no no no no no no no no

## 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

## no no no no no no no no no no no no no no yes no

## 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48

## yes no yes no no no no no no no no no no no no no

## 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64

## no no no no no no no no yes no yes no no no no no

## 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

## no no no no yes no no no no yes no no no no no yes

## 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96

## no no no no no no no no no no no no no no no no

## 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112

## yes no no yes no no no no no no no no no yes no no

## 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128

## no no no no no no no no yes yes no no no no no no

## 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144

## no no yes no no no yes no no no no no no yes no no

## 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160

## yes no yes no no no no no no no no no no no no no

## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176

## no no no no no no no no no no no no yes no no yes

## 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192

## no no no no no no no no no no no no no no no no

## 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208

## no no no no no no no no yes yes no no no no no no

## 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224

## no no no no no no no no no no no yes no no no no

## 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240

## yes no no no no no no no no no no no no no no no

## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256

## no no no yes no no no no no no no no no no no no

## 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272

## no no no no no no yes no no no yes no no no no no

## 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288

## no no no no no no no no no no no no no no no no

## 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304

## no no yes no no no no no no no no no yes no yes no

## 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320

## no no no no no no no no no no no no no no no no

## 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336

## no no no no no no no no no no no yes no no no no

## 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352

## no no no no no yes no no no no no no no no yes no

## 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368

## no no no yes no no no no no no no no no no no no

## 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384

## no no no no no no no no no no no no no no no no

## 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400

## no no no no no no no no no no no no no no no no

## 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416

## yes no no yes no no no no no no no no no no no no

## 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432

## no no no no no no no no no no no no no no no no

## 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448

## no no no no no no no no no no no no no no no no

## 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464

## no no no no no no no no yes no no no no no no no

## 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480

## no no no no no no no no yes no no no no no no no

## 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496

## no no no yes no no no no no no no no no no no no

## 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512

## no no no no no no no no no no no no yes no no no

## 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528

## no yes no no no no no no no no no no no no no no

## 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544

## no no no no no no no no no yes no no no no no no

## 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560

## no no no yes no no no no no no no no no no no yes

## 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576

## no no no no no no no no no no no no no no no no

## 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592

## no yes no no no no no no no no no no no no yes no

## 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608

## no no no no no no yes no no no yes no no no no no

## 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624

## no no yes no no no no no no no no no no no no yes

## 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640

## no no no no no no no no yes no yes no no no no no

## 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656

## no no no no no no no no no no no no no no yes no

## 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672

## no no no no no no no no no no no no no no yes no

## 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688

## no no no no no no no no no no no no no yes no no

## 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704

## no no no no no no no no no no no no no no no no

## 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720

## no no no no no no no no yes no no yes no no no no

## 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736

## no no no yes no no no no no no no no no yes yes no

## 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752

## no no no yes no no no no no no no no no no no no

## 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768

## yes no no no no no no no no no no no no no no no

## 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784

## no no no no yes no no no no no no no yes no no no

## 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800

## no no yes no no no no yes no no no no no no no no

## 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816

## no no no no yes no no no no no no no no no no no

## 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832

## no no no no no no no yes no no no yes no no no yes

## 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848

## no no no no no yes no no yes yes no no no no no no

## 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864

## no no no no no no no no no yes no no no no no no

## 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880

## no no no no no no no no yes no no no no no yes no

## 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896

## no no no no no no no no no no no no no no no no

## 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912

## no no no no yes no no no no no yes no no yes no no

## 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928

## no no no no no yes no no no no no no no no no no

## 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944

## no no no no yes no no no no no no no no no no no

## 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960

## no no no yes no no no no no no no no no no no no

## 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976

## no yes no no no no no no no no no no no no no yes

## 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992

## no no yes no no yes no no no no no no no no no no

## 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008

## no no no no no no no yes no no no no no yes no yes

## 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024

## no no yes no no no no no no no no no no no no no

## 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040

## no no no no no no no no no no no no no no no no

## 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056

## no no no no no no no no no no no no no no no no

## 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072

## no no no no no no no no no no no no no no no no

## 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088

## no no no no no no yes no no no no no no no no no

## 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104

## no no yes no no no no no no no no no no no yes no

## 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120

## no no yes no no no no no no yes no no no no no no

## 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136

## no no no no no no no no no no no no no no yes no

## 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152

## no no no no no yes no no no no no no no no no no

## 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168

## no no no no no no no no no no yes no no no no no

## 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184

## no no no no yes no no no no no no no no no no no

## 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200

## no no no yes no no no no no yes no no no no no no

## 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216

## no yes no no no yes no no no no no yes no no yes no

## 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232

## no no no no no no no no no no yes no no no no no

## 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248

## no no yes yes no no yes no no no no no no no no no

## 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264

## no yes no no no no no no no no no no no no no no

## 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280

## no no yes no no no no no no no no no no no no no

## 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296

## yes no no no no no yes no no no no no no yes no no

## 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312

## no no yes no no yes no no no no no no yes no no no

## 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328

## no yes no no no no no no no no no no no no no no

## 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344

## yes no no no no no no no no no no no no no no no

## 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360

## no no no no yes no no no no no no no no no no no

## 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376

## no yes no no no no no no no no no no no no no no

## 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392

## no no no no no no no no yes no no no no no yes yes

## 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408

## no no no no no no no no no no no no no no no no

## 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424

## no no no no yes yes no yes no no no no no no no no

## 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440

## yes no no no no no no yes no no no no no yes no no

## 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456

## no no no no yes yes no no no no no no no no no no

## 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472

## no no no no no no no no no no no no no no no no

## 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488

## no no no no no no no no no no no no no no yes no

## 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504

## no no no no yes no no no no no no no yes no no no

## 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520

## no yes no no no yes no no no no no no no no no no

## 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536

## no no no no yes no no no yes no no no no no no no

## 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552

## no yes no no yes no no no yes no no no no no no no

## 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568

## yes no no no no no no no no no no no no no yes no

## 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584

## no no yes no yes no no no no no no no no no no no

## 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600

## no no no no no no no no no no no no no no no no

## Levels: no yes

*#Displaying the count of Yes/No Present in Churn\_Probability Column.*

table(Customers\_To\_Predict$Churn\_Probability)

##

## no yes

## 1453 147

**Interpretation**

As we took the 25 % test data, we are having 1600 customers in test data and we can perform the forecast future churn on them.

the results are:

1453 customers are not ready to move out of ABC wireless network.

147 customers moving from ABC wireless to another network.