Interview Attendance Prediction

# Include Libraries

library(tidyverse)

## -- Attaching packages --------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.0.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.6  
## v tidyr 0.8.1 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(stringr)  
library(dplyr)  
library(data.table)

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

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

library(boot)  
require(ISLR)

## Loading required package: ISLR

Read File

# Read csv  
int\_csv <- read.csv("c:/Data Science/Final Project/Interview.csv")  
int\_df <- data.frame(int\_csv)  
str(int\_df)

## 'data.frame': 1233 obs. of 26 variables:  
## $ Client.name : Factor w/ 15 levels "ANZ","Aon Hewitt",..: 8 8 8 8 8 2 2 2 2 2 ...  
## $ Industry : Factor w/ 7 levels "BFSI","Electronics",..: 6 6 6 6 6 5 5 5 5 5 ...  
## $ Location : Factor w/ 11 levels "Bangalore","chennai",..: 3 3 3 3 3 8 8 8 8 8 ...  
## $ Position.to.be.closed : Factor w/ 7 levels "AML","Dot Net",..: 4 4 4 4 4 6 6 6 6 6 ...  
## $ Nature.of.Skillset : Factor w/ 92 levels "#NAME?","10.00 AM",..: 74 74 74 74 74 74 74 74 74 74 ...  
## $ Interview.Type : Factor w/ 6 levels "Sceduled walkin",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ Name.Cand.ID. : Factor w/ 1233 levels "Candidate 1",..: 1 346 457 568 679 790 901 1012 1123 2 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 1 2 1 ...  
## $ Candidate.Current.Location : Factor w/ 10 levels "Bangalore","chennai",..: 3 3 3 3 3 8 8 8 8 8 ...  
## $ Candidate.Job.Location : Factor w/ 7 levels "Bangalore","Chennai",..: 5 1 2 2 1 4 4 4 4 4 ...  
## $ Interview.Venue : Factor w/ 7 levels "Bangalore","Chennai",..: 5 5 5 5 5 4 4 4 4 4 ...  
## $ Candidate.Native.location : Factor w/ 45 levels "Agra","Ahmedabad",..: 22 39 11 11 11 20 20 31 17 17 ...  
## $ Have.you.obtained.the.necessary.permission.to.start.at.the.required.time : Factor w/ 7 levels "Na","No","NO",..: 6 6 NA 6 6 6 6 6 6 6 ...  
## $ Hope.there.will.be.no.unscheduled.meetings : Factor w/ 7 levels "cant Say","Na",..: 7 7 2 7 7 7 7 7 7 7 ...  
## $ Can.I.Call.you.three.hours.before.the.interview.and.follow.up.on.your.attendance.for.the.interview: Factor w/ 5 levels "Na","No","No Dont",..: 5 5 NA 2 5 5 5 5 5 5 ...  
## $ Can.I.have.an.alternative.number..desk.number..I.assure.you.that.I.will.not.trouble.you.too.much : Factor w/ 6 levels "na","Na","No",..: 6 6 NA 6 3 6 6 6 6 6 ...  
## $ Have.you.taken.a.printout.of.your.updated.resume..Have.you.read.the.JD.and.understood.the.same : Factor w/ 8 levels "na","Na","No",..: 8 8 NA 3 8 8 8 8 8 8 ...  
## $ Are.you.clear.with.the.venue.details.and.the.landmark. : Factor w/ 7 levels "na","Na","no",..: 7 7 NA 7 7 7 7 7 7 7 ...  
## $ Has.the.call.letter.been.shared : Factor w/ 12 levels "Havent Checked",..: 11 11 NA 11 11 11 11 11 11 11 ...  
## $ Expected.Attendance : Factor w/ 7 levels "10.30 Am","11:00 AM",..: 7 7 5 5 5 7 7 7 7 7 ...  
## $ Observed.Attendance : Factor w/ 8 levels "no","No","NO",..: 2 2 2 2 2 7 7 7 7 2 ...  
## $ Marital.Status : Factor w/ 2 levels "Married","Single": 2 2 2 2 1 2 2 2 2 2 ...  
## $ X : logi NA NA NA NA NA NA ...  
## $ X.1 : logi NA NA NA NA NA NA ...  
## $ X.2 : logi NA NA NA NA NA NA ...  
## $ X.3 : logi NA NA NA NA NA NA ...

Remove Unwanted columns

# Select the columns required for Analysis  
int\_df <- subset(int\_df, select = -c(2,3, 4, 5, 6, 7, 23, 24, 25, 26 ))  
names(int\_df)

## [1] "Client.name"   
## [2] "Gender"   
## [3] "Candidate.Current.Location"   
## [4] "Candidate.Job.Location"   
## [5] "Interview.Venue"   
## [6] "Candidate.Native.location"   
## [7] "Have.you.obtained.the.necessary.permission.to.start.at.the.required.time"   
## [8] "Hope.there.will.be.no.unscheduled.meetings"   
## [9] "Can.I.Call.you.three.hours.before.the.interview.and.follow.up.on.your.attendance.for.the.interview"  
## [10] "Can.I.have.an.alternative.number..desk.number..I.assure.you.that.I.will.not.trouble.you.too.much"   
## [11] "Have.you.taken.a.printout.of.your.updated.resume..Have.you.read.the.JD.and.understood.the.same"   
## [12] "Are.you.clear.with.the.venue.details.and.the.landmark."   
## [13] "Has.the.call.letter.been.shared"   
## [14] "Expected.Attendance"   
## [15] "Observed.Attendance"   
## [16] "Marital.Status"

str(int\_df)

## 'data.frame': 1233 obs. of 16 variables:  
## $ Client.name : Factor w/ 15 levels "ANZ","Aon Hewitt",..: 8 8 8 8 8 2 2 2 2 2 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 1 2 1 ...  
## $ Candidate.Current.Location : Factor w/ 10 levels "Bangalore","chennai",..: 3 3 3 3 3 8 8 8 8 8 ...  
## $ Candidate.Job.Location : Factor w/ 7 levels "Bangalore","Chennai",..: 5 1 2 2 1 4 4 4 4 4 ...  
## $ Interview.Venue : Factor w/ 7 levels "Bangalore","Chennai",..: 5 5 5 5 5 4 4 4 4 4 ...  
## $ Candidate.Native.location : Factor w/ 45 levels "Agra","Ahmedabad",..: 22 39 11 11 11 20 20 31 17 17 ...  
## $ Have.you.obtained.the.necessary.permission.to.start.at.the.required.time : Factor w/ 7 levels "Na","No","NO",..: 6 6 NA 6 6 6 6 6 6 6 ...  
## $ Hope.there.will.be.no.unscheduled.meetings : Factor w/ 7 levels "cant Say","Na",..: 7 7 2 7 7 7 7 7 7 7 ...  
## $ Can.I.Call.you.three.hours.before.the.interview.and.follow.up.on.your.attendance.for.the.interview: Factor w/ 5 levels "Na","No","No Dont",..: 5 5 NA 2 5 5 5 5 5 5 ...  
## $ Can.I.have.an.alternative.number..desk.number..I.assure.you.that.I.will.not.trouble.you.too.much : Factor w/ 6 levels "na","Na","No",..: 6 6 NA 6 3 6 6 6 6 6 ...  
## $ Have.you.taken.a.printout.of.your.updated.resume..Have.you.read.the.JD.and.understood.the.same : Factor w/ 8 levels "na","Na","No",..: 8 8 NA 3 8 8 8 8 8 8 ...  
## $ Are.you.clear.with.the.venue.details.and.the.landmark. : Factor w/ 7 levels "na","Na","no",..: 7 7 NA 7 7 7 7 7 7 7 ...  
## $ Has.the.call.letter.been.shared : Factor w/ 12 levels "Havent Checked",..: 11 11 NA 11 11 11 11 11 11 11 ...  
## $ Expected.Attendance : Factor w/ 7 levels "10.30 Am","11:00 AM",..: 7 7 5 5 5 7 7 7 7 7 ...  
## $ Observed.Attendance : Factor w/ 8 levels "no","No","NO",..: 2 2 2 2 2 7 7 7 7 2 ...  
## $ Marital.Status : Factor w/ 2 levels "Married","Single": 2 2 2 2 1 2 2 2 2 2 ...

Rename column names to make it easy

setnames(int\_df, old=c("Candidate.Current.Location","Candidate.Job.Location", "Candidate.Native.location", "Interview.Venue"), new=c("curr\_Location", "Job\_Location", "Native", "Venue"))  
setnames(int\_df, old=c("Have.you.obtained.the.necessary.permission.to.start.at.the.required.time"), new=c("Permission"))  
setnames(int\_df, old=c("Hope.there.will.be.no.unscheduled.meetings", "Can.I.Call.you.three.hours.before.the.interview.and.follow.up.on.your.attendance.for.the.interview", "Can.I.have.an.alternative.number..desk.number..I.assure.you.that.I.will.not.trouble.you.too.much", "Have.you.taken.a.printout.of.your.updated.resume..Have.you.read.the.JD.and.understood.the.same", "Are.you.clear.with.the.venue.details.and.the.landmark.", "Has.the.call.letter.been.shared"), new=c("Unscheduled\_Meetings", "Call\_before\_3hrs", "Alt\_no", "Resume\_prinout", "Landmark", "Call\_letter"))  
setnames(int\_df, old=c("Expected.Attendance"), new=c("Rec\_Expectation"))  
names(int\_df)

## [1] "Client.name" "Gender" "curr\_Location"   
## [4] "Job\_Location" "Venue" "Native"   
## [7] "Permission" "Unscheduled\_Meetings" "Call\_before\_3hrs"   
## [10] "Alt\_no" "Resume\_prinout" "Landmark"   
## [13] "Call\_letter" "Rec\_Expectation" "Observed.Attendance"   
## [16] "Marital.Status"

Checking each column for unique values

Client Name:

unique(int\_df$Client.name)

## [1] Hospira Aon Hewitt   
## [3] UST Standard Chartered Bank   
## [5] ANZ Pfizer   
## [7] Standard Chartered Bank Chennai Aon hewitt Gurgaon   
## [9] Astrazeneca Flextronics   
## [11] Prodapt Williams Lea   
## [13] Barclays Hewitt   
## [15] Woori Bank   
## 15 Levels: ANZ Aon Hewitt Aon hewitt Gurgaon Astrazeneca ... Woori Bank

Too many names for the same Client - Data Error Data Correction using “R” string features Correct “Aon hewitt Gurgaon” to “Hewitt”

int\_df$Client.name<-str\_replace(int\_df$Client.name, "Aon hewitt Gurgaon", "Hewitt")

correct “Aon Hewitt” to “Hewitt”

int\_df$Client.name<-str\_replace(int\_df$Client.name, "Aon Hewitt", "Hewitt")

correct “Standard Chartered Bank Chennai” to “Standard Chartered Bank”

int\_df$Client.name<-str\_replace(int\_df$Client.name, "Standard Chartered Bank Chennai", "Standard Chartered Bank")

Check Current Location column

unique(int\_df$curr\_Location)

## [1] Chennai Gurgaon Bangalore Hyderabad Delhi chennai Cochin   
## [8] Noida CHENNAI chennai   
## 10 Levels: Bangalore chennai Chennai CHENNAI chennai Cochin ... Noida

# Tidy - trim spaces  
int\_df$curr\_Location <- str\_trim(int\_df$curr\_Location)  
  
#data correct "Gurgaonr" , "Chennai"  
int\_df$curr\_Location <- str\_replace(int\_df$curr\_Location, "Gurgaonr", "Gurgaon")  
int\_df$curr\_Location <- str\_replace(int\_df$curr\_Location, "chennai", "Chennai")  
int\_df$curr\_Location <- str\_replace(int\_df$curr\_Location, "CHENNAI", "Chennai")  
unique(int\_df$curr\_Location)

## [1] "Chennai" "Gurgaon" "Bangalore" "Hyderabad" "Delhi" "Cochin"   
## [7] "Noida"

Check Job\_Location column

unique(int\_df$Job\_Location)

## [1] Hosur Bangalore Chennai Gurgaon Visakapatinam  
## [6] Cochin Noida   
## Levels: Bangalore Chennai Cochin Gurgaon Hosur Noida Visakapatinam

# Remove Unwanted Spaces - trim padding  
int\_df$Job\_Location <- str\_trim(int\_df$Job\_Location)

Check Venue

unique(int\_df$Venue)

## [1] Hosur Gurgaon Bangalore Chennai Hyderabad Cochin Noida   
## Levels: Bangalore Chennai Cochin Gurgaon Hosur Hyderabad Noida

# trim padding  
int\_df$Venue <- str\_trim(int\_df$Venue)

Check Native

unique(int\_df$Native)

## [1] Hosur Trichy Chennai Gurgaon Noida   
## [6] Delhi /NCR Cochin Trivandrum Bangalore Coimbatore   
## [11] Salem Tanjore Hyderabad Mumbai Pune   
## [16] Kolkata Allahabad Panjim Cuttack Visakapatinam  
## [21] Belgaum Patna Chitoor Anantapur Warangal   
## [26] Ahmedabad Kurnool Vijayawada Vellore Pondicherry   
## [31] Nagercoil Agra Bhubaneshwar Ghaziabad Baddi   
## [36] Tuticorin Tirupati Faizabad Ambur Chandigarh   
## [41] Mysore Hissar Delhi Kanpur Lucknow   
## 45 Levels: Agra Ahmedabad Allahabad Ambur Anantapur Baddi ... Warangal

# Remove spaces - trim padding  
int\_df$Native <- str\_trim(int\_df$Native)  
unique(int\_df$Native)

## [1] "Hosur" "Trichy" "Chennai" "Gurgaon"   
## [5] "Noida" "Delhi /NCR" "Cochin" "Trivandrum"   
## [9] "Bangalore" "Coimbatore" "Salem" "Tanjore"   
## [13] "Hyderabad" "Mumbai" "Pune" "Kolkata"   
## [17] "Allahabad" "Panjim" "Cuttack" "Visakapatinam"  
## [21] "Belgaum" "Patna" "Chitoor" "Anantapur"   
## [25] "Warangal" "Ahmedabad" "Kurnool" "Vijayawada"   
## [29] "Vellore" "Pondicherry" "Nagercoil" "Agra"   
## [33] "Bhubaneshwar" "Ghaziabad" "Baddi" "Tuticorin"   
## [37] "Tirupati" "Faizabad" "Ambur" "Chandigarh"   
## [41] "Mysore" "Hissar" "Delhi" "Kanpur"   
## [45] "Lucknow"

attach(int\_df)

I converted the data as scores, to be used to predict the outcome.

The Data is used to predict if the Candidate will attend the Job Interview or not and to see how many will turn out for the Interview. The columns Current Location(where the Candidate is living), Job Location(where the JOb is located), Venue(where the Interview will be held), Native(Native of the Candidate) were used as indicators. As a general Psychology, many want to stay in their Native. If they are currently working in a different location,they might be interested in a Job Change and would want to attend the interview.

I have created 2 type of Scores - 1 for Logistic Regression Model and another to be be used in Decision Tree Model. Based on the condition, if the likelihood for attending is more, then a ‘1’ or “Yes” is given for score. If the likelihood for attending is less, then a ‘0’ or “No” is given for the score.

Score based on Location:

Loc\_score1, Loc\_tree1 - Based on the Assumption, When the Current Location is not equal to native, the candidate is likely to go for a job change.

# Loc\_score1 -   
for (i in 1:nrow(int\_df))  
{  
 if (Native[i] != curr\_Location[i])   
 int\_df$Loc\_score1[i] <- 1  
 else  
 int\_df$Loc\_score1[i] <- 0  
}  
table(int\_df$Loc\_score1)

##   
## 0 1   
## 766 467

# Loc\_tree1 - Current Location not equal to native  
for (i in 1:length(Native))  
{  
 if (Native[i] != curr\_Location[i])   
 int\_df$Loc\_tree1[i] <- "Yes"  
 else  
 int\_df$Loc\_tree1[i] <- "No"  
}  
table(int\_df$Loc\_tree1)

##   
## No Yes   
## 766 467

Loc\_score2, Loc\_tree2 - Based on the Assumption, when the Native of the candidate is equal to Job Location, the candidate is likely to go for a Job Change.

for (i in 1:length(int\_df$Native))  
{  
 if ((int\_df$Native[i] == int\_df$Job\_Location[i]) )  
 int\_df$Loc\_score2[i] <- 1  
 else  
 int\_df$Loc\_score2[i] <- 0  
}  
table(int\_df$Loc\_score2)

##   
## 0 1   
## 478 755

for (i in 1:length(int\_df$Native))  
{  
 if ((int\_df$Native[i] == int\_df$Job\_Location[i]) )  
 int\_df$Loc\_tree2[i] <- "Yes"  
 else  
 int\_df$Loc\_tree2[i] <- "No"  
}  
table(int\_df$Loc\_tree2)

##   
## No Yes   
## 478 755

Loc\_score3, Loc\_tree3 - Based on the Assumption, when the Native is the same as Venue, the candidate is likely to go for a Job Change.

for (i in 1:length(int\_df$Native))  
{  
 if (int\_df$Native[i] == int\_df$Venue[i])  
 int\_df$Loc\_score3[i] <- 1  
 else  
 int\_df$Loc\_score3[i] <- 0  
}  
table(int\_df$Loc\_score3)

##   
## 0 1   
## 461 772

for (i in 1:length(int\_df$Native))  
{  
 if (int\_df$Native[i] == int\_df$Venue[i])  
 int\_df$Loc\_tree3[i] <- "Yes"  
 else  
 int\_df$Loc\_tree3[i] <- "No"  
}  
table(int\_df$Loc\_tree3)

##   
## No Yes   
## 461 772

The Recruiter checks the Readiness of the Candidate, by asking several questions. These questions were posed to check the Readiness, interest of the Candidate.The more Ready, interested, the more likelihood in attending the interview.

From the question, “Have you obtained the necessary permission to start at the required time”, one can understand how much the Candidate is ready for a change.

From my point of view, Yes, NA - shows more interest, may attend interview, Not yet, No, Yet to confirm, nulls - may not attend interview

int\_df$Permission <- str\_trim(int\_df$Permission)  
  
  
int\_df <- mutate(int\_df, Perm\_score =  
 ifelse(grepl("Yes", int\_df$Permission), 1,  
 ifelse(grepl("yes", int\_df$Permission), 1,  
 ifelse(grepl("Na", int\_df$Permission), 1, 0   
 ))))  
table(int\_df$Perm\_score)

##   
## 0 1   
## 307 926

# For Decision Tree  
int\_df <- mutate(int\_df, Perm\_tree =  
 ifelse(grepl("Yes", int\_df$Permission), "Yes",  
 ifelse(grepl("yes", int\_df$Permission), "Yes",  
 ifelse(grepl("Na", int\_df$Permission), "Yes", "No"   
 ))))  
table(int\_df$Perm\_tree)

##   
## No Yes   
## 307 926

For the question, “Hope there will be no unscheduled meetings”, answers such as “Yes”, “Na” are considered as “may attend”, “NO”, “Not Sure”, “Cant say” can be considered as “may not attend”

unique(Unscheduled\_Meetings)

## [1] Yes Na No <NA> yes Not Sure cant Say Not sure  
## Levels: cant Say Na No Not sure Not Sure yes Yes

int\_df <- mutate(int\_df, Uns\_meeting\_score =  
 ifelse(grepl("Yes", int\_df$Unscheduled\_Meetings), 1,  
 ifelse(grepl("No", int\_df$Unscheduled\_Meetings), 0,  
 ifelse(grepl("Not Sure", int\_df$Unscheduled\_Meetings), 0,  
 ifelse(grepl("Not sure", int\_df$Unscheduled\_Meetings), 0,  
 ifelse(grepl("cant Say", int\_df$Unscheduled\_Meetings), 0,  
 ifelse(grepl("yes", int\_df$Unscheduled\_Meetings), 1,  
 ifelse(grepl("Na", int\_df$Unscheduled\_Meetings), 1, 0  
 )  
 )))))))  
table(int\_df$Uns\_meeting\_score)

##   
## 0 1   
## 259 974

For Decision Tree

int\_df <- mutate(int\_df, Uns\_meet\_tree =  
 ifelse(grepl("Yes", int\_df$Unscheduled\_Meetings), "Yes",  
 ifelse(grepl("No", int\_df$Unscheduled\_Meetings), "No",  
 ifelse(grepl("Not Sure", int\_df$Unscheduled\_Meetings), "Not Sure",  
 ifelse(grepl("Not sure", int\_df$Unscheduled\_Meetings), "Not Sure",  
 ifelse(grepl("cant Say", int\_df$Unscheduled\_Meetings), "Not Sure",  
 ifelse(grepl("yes", int\_df$Unscheduled\_Meetings), "Yes",  
 ifelse(grepl("Na", int\_df$Unscheduled\_Meetings), "Yes", "No"  
 )  
 )))))))  
table(int\_df$Uns\_meet\_tree)

##   
## No Not Sure Yes   
## 258 1 974

Question: “Can I Call you three hours before the interview and follow up on your attendance for the interview?” Answers : “Yes”, “Na” - considered for “may attend” : “No”, nulls, “Dont” - considered “may not attend”

unique(Call\_before\_3hrs)

## [1] Yes <NA> No No Dont Na yes   
## Levels: Na No No Dont yes Yes

int\_df$Call\_before\_3hrs <- str\_trim(int\_df$Call\_before\_3hrs)  
  
int\_df <- mutate(int\_df, Call\_score =  
 ifelse(grepl("Yes", int\_df$Call\_before\_3hrs), 1,  
 ifelse(grepl("No", int\_df$Call\_before\_3hrs), 0,  
 ifelse(grepl("No", int\_df$Call\_before\_3hrs), 0,  
 ifelse(grepl("Dont", int\_df$Call\_before\_3hrs), 0,   
 ifelse(grepl("yes", int\_df$Call\_before\_3hrs), 1,  
 ifelse(grepl("Na", int\_df$Call\_before\_3hrs), 1, 0)))))))  
table(int\_df$Call\_score)

##   
## 0 1   
## 258 975

# For Decision Tree  
  
int\_df <- mutate(int\_df, Call\_tree =  
 ifelse(grepl("Yes", int\_df$Call\_before\_3hrs), "Yes",  
 ifelse(grepl("No", int\_df$Call\_before\_3hrs), "No",  
 ifelse(grepl("No", int\_df$Call\_before\_3hrs), "No",  
 ifelse(grepl("Dont", int\_df$Call\_before\_3hrs), "No",   
 ifelse(grepl("yes", int\_df$Call\_before\_3hrs), "Yes",  
 ifelse(grepl("Na", int\_df$Call\_before\_3hrs), "Yes", "No")))))))  
table(int\_df$Call\_tree)

##   
## No Yes   
## 258 975

Question : Can I have an alternative number/ desk number?  
Answer : “Yes”, “na” : “No”, nulls

unique(Alt\_no)

## [1] Yes <NA>   
## [3] No No I have only thi number  
## [5] na yes   
## [7] Na   
## Levels: na Na No No I have only thi number yes Yes

int\_df$Alt\_no <- str\_trim(int\_df$Alt\_no)  
int\_df <- mutate(int\_df, Alt\_score =  
 ifelse(grepl("Yes", int\_df$Alt\_no), 1,  
 ifelse(grepl("No", int\_df$Alt\_no), 0,  
 ifelse(grepl("na", int\_df$Alt\_no), 1,  
 ifelse(grepl("yes", int\_df$Alt\_no), 1,  
 ifelse(grepl("Na", int\_df$Alt\_no), 1, 0))))))  
table(int\_df$Alt\_score)

##   
## 0 1   
## 276 957

#Alt\_tree  
int\_df <- mutate(int\_df, Alt\_tree =  
 ifelse(grepl("Yes", int\_df$Alt\_no), "Yes",  
 ifelse(grepl("No", int\_df$Alt\_no), "No",  
 ifelse(grepl("na", int\_df$Alt\_no), "Yes",  
 ifelse(grepl("yes", int\_df$Alt\_no), "Yes",  
 ifelse(grepl("Na", int\_df$Alt\_no), "Yes", "No"))))))  
table(int\_df$Alt\_tree)

##   
## No Yes   
## 276 957

Question : “Have you taken a printout of your updated resume?” Answer : “Yes”, “Na” - considered “may attend” : “No”, “Not yet, nulls” - considerd “may not attend”

unique(Resume\_prinout)

## [1] Yes <NA> No   
## [4] No- will take it soon Not yet na   
## [7] yes Na Not Yet   
## Levels: na Na No No- will take it soon Not yet Not Yet yes Yes

int\_df$Resume\_prinout <- str\_trim(int\_df$Resume\_prinout)  
int\_df <- mutate(int\_df, Res\_score =  
 ifelse(grepl("Yes", int\_df$Resume\_prinout), 1,  
 ifelse(grepl("No", int\_df$Resume\_prinout), 0,  
 ifelse(grepl("Not yet", int\_df$Resume\_prinout), 0,  
 ifelse(grepl("Not Yet", int\_df$Resume\_prinout), 0,  
 ifelse(grepl("na", int\_df$Resume\_prinout), 1,  
 ifelse(grepl("Na", int\_df$Resume\_prinout), 1,  
 ifelse(grepl("yes", int\_df$Resume\_prinout), 1, 0))))))))  
table(int\_df$Res\_score)

##   
## 0 1   
## 271 962

# Resume\_tree  
int\_df <- mutate(int\_df, Res\_tree =  
 ifelse(grepl("Yes", int\_df$Resume\_prinout), "Yes",  
 ifelse(grepl("No", int\_df$Resume\_prinout), "No",  
 ifelse(grepl("Not yet", int\_df$Resume\_prinout), "No",  
 ifelse(grepl("Not Yet", int\_df$Resume\_prinout), "No",  
 ifelse(grepl("na", int\_df$Resume\_prinout), "Uncertain",  
 ifelse(grepl("Na", int\_df$Resume\_prinout), "Uncertain",  
 ifelse(grepl("yes", int\_df$Resume\_prinout), "Yes", "No"))))))))  
table(int\_df$Res\_tree)

##   
## No Uncertain Yes   
## 271 20 942

Question : Are you clear with the venue details and the landmark? Ans : Yes, NA - considered “May Attend” : No, nulls - considered “May not Attend”

int\_df$Landmark <- str\_trim(int\_df$Landmark)  
unique(Landmark)

## [1] Yes <NA> No   
## [4] No- I need to check na yes   
## [7] Na no   
## Levels: na Na no No No- I need to check yes Yes

int\_df <- mutate(int\_df, Land\_score =  
 ifelse(grepl("Yes", int\_df$Landmark), 1,  
 ifelse(grepl("No", int\_df$Landmark), 0,  
 ifelse(grepl("na", int\_df$Landmark), 1,  
 ifelse(grepl("Na", int\_df$Landmark), 1,  
 ifelse(grepl("yes", int\_df$Landmark), 1, 0))))))  
  
table(int\_df$Land\_score)

##   
## 0 1   
## 265 968

# Land\_tree  
  
int\_df <- mutate(int\_df, Land\_tree =  
 ifelse(grepl("Yes", int\_df$Landmark), "Yes",  
 ifelse(grepl("No", int\_df$Landmark), "No",  
 ifelse(grepl("na", int\_df$Landmark), "Uncertain",  
 ifelse(grepl("Na", int\_df$Landmark), "Uncertain",  
 ifelse(grepl("yes", int\_df$Landmark), "Yes", "No"))))))  
  
table(int\_df$Land\_tree)

##   
## No Uncertain Yes   
## 265 20 948

Question : Has the call letter been shared? Answer : “Yes”, “NA” - considered “May Attend” : “No”, Null, others - considered “may not Attend”

int\_df$Call\_letter <- str\_trim(int\_df$Call\_letter)  
unique(Call\_letter)

## [1] Yes <NA> Havent Checked No   
## [5] Need To Check Not sure Yet to Check Not Sure   
## [9] Not yet no na yes   
## [13] Na   
## 12 Levels: Havent Checked na Na Need To Check no No Not sure ... Yet to Check

int\_df <- mutate(int\_df, Letter\_score =  
 ifelse(grepl("Yes", int\_df$Call\_letter), 1,  
 ifelse(grepl("na", int\_df$Call\_letter), 1,  
 ifelse(grepl("Na", int\_df$Call\_letter), 1,  
 ifelse(grepl("yes", int\_df$Call\_letter), 1, 0)))))  
  
table(int\_df$Letter\_score)

##   
## 0 1   
## 279 954

# Letter\_tree  
int\_df <- mutate(int\_df, Letter\_tree =  
 ifelse(grepl("Yes", int\_df$Call\_letter), "Yes",  
 ifelse(grepl("na", int\_df$Call\_letter), "Uncertain",  
 ifelse(grepl("Na", int\_df$Call\_letter), "Uncertain",  
 ifelse(grepl("yes", int\_df$Call\_letter), "Yes", "No")))))  
  
table(int\_df$Letter\_tree)

##   
## No Uncertain Yes   
## 279 20 934

Predictors: Observed Attendance

unique(int\_df$Observed.Attendance)

## [1] No Yes yes no yes No NO no   
## Levels: no No NO no No yes Yes yes

int\_df$Observed.Attendance <- str\_trim(int\_df$Observed.Attendance)  
  
# For Classification Model  
int\_df <- mutate(int\_df, Obs\_Att\_class =  
 ifelse(grepl("Yes", int\_df$Observed.Attendance), 1,  
 ifelse(grepl("yes", int\_df$Observed.Attendance), 1,   
 ifelse(grepl(" yes", int\_df$Observed.Attendance), 1, 0))))  
table(int\_df$Obs\_Att\_class)

##   
## 0 1   
## 450 783

# For Decision Tree  
int\_df <- mutate(int\_df, Obs\_Att\_tree =  
 ifelse(grepl("Yes", Observed.Attendance), "Yes",  
 ifelse(grepl("yes", Observed.Attendance), "Yes",   
 ifelse(grepl(" yes", Observed.Attendance), "Yes", "No"))))  
table(int\_df$Obs\_Att\_tree)

##   
## No Yes   
## 450 783

Logistic Regression using All variables - Train and Test Dataset, Splitting into 75:25

attach(int\_df)

## The following objects are masked from int\_df (pos = 4):  
##   
## Alt\_no, Call\_before\_3hrs, Call\_letter, Client.name,  
## curr\_Location, Gender, Job\_Location, Landmark, Marital.Status,  
## Native, Observed.Attendance, Permission, Rec\_Expectation,  
## Resume\_prinout, Unscheduled\_Meetings, Venue

set.seed(1)  
train = sample(1233, 900)  
  
glm.fit = glm(Obs\_Att\_class ~ Loc\_score1 + Loc\_score2 + Loc\_score3 +  
 Perm\_score + Call\_score + Uns\_meeting\_score + Alt\_score +   
 Res\_score + Land\_score + Letter\_score + Marital.Status + Gender,  
 data=int\_df, family = binomial, subset = train)

Predict on the train Dataset:

glm.probs=predict(glm.fit,type="response")  
summary(glm.probs)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.06741 0.55522 0.71519 0.63333 0.73700 0.96122

glm.pred = rep(0, 900)  
glm.pred = ifelse(glm.probs > 0.5,1,0)  
str(glm.pred)

## Named num [1:900] 1 0 1 1 0 1 1 1 1 1 ...  
## - attr(\*, "names")= chr [1:900] "328" "459" "706" "1118" ...

str(Obs\_Att\_class[train])

## num [1:900] 1 1 1 1 0 1 1 1 0 1 ...

table(glm.pred, Obs\_Att\_class[train])

##   
## glm.pred 0 1  
## 0 144 74  
## 1 186 496

mean(glm.pred == Obs\_Att\_class[train])

## [1] 0.7111111

## Train Error rate: 100 - 71.11 = 29.89%

Predict on Test Data set:

glm.probs=predict(glm.fit,newdata=int\_df[-train,],type="response")  
summary(glm.probs)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.06872 0.54641 0.71519 0.63025 0.73700 0.94635

glm.pred = rep(0, 333)  
glm.pred = ifelse(glm.probs > 0.5,1,0)  
  
table(glm.pred, Obs\_Att\_class[-train])

##   
## glm.pred 0 1  
## 0 46 36  
## 1 74 177

mean(glm.pred == Obs\_Att\_class[-train])

## [1] 0.6696697

## Test Error Rate: 100 - 66.97 = 33.03%

summary(glm.fit)

##   
## Call:  
## glm(formula = Obs\_Att\_class ~ Loc\_score1 + Loc\_score2 + Loc\_score3 +   
## Perm\_score + Call\_score + Uns\_meeting\_score + Alt\_score +   
## Res\_score + Land\_score + Letter\_score + Marital.Status +   
## Gender, family = binomial, data = int\_df, subset = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2148 -0.9510 0.7193 0.8188 1.9175   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.8312 0.8544 -0.973 0.3307   
## Loc\_score1 0.4812 0.8282 0.581 0.5612   
## Loc\_score2 -1.0444 0.4754 -2.197 0.0280 \*   
## Loc\_score3 1.2263 0.8964 1.368 0.1713   
## Perm\_score 2.3704 0.4514 5.251 1.51e-07 \*\*\*  
## Call\_score -1.9495 1.1420 -1.707 0.0878 .   
## Uns\_meeting\_score 0.2786 0.8025 0.347 0.7285   
## Alt\_score 0.5076 0.5924 0.857 0.3915   
## Res\_score 0.3641 0.9121 0.399 0.6898   
## Land\_score -0.4965 0.7372 -0.674 0.5006   
## Letter\_score 0.5147 0.6347 0.811 0.4174   
## Marital.StatusSingle 0.1895 0.1596 1.188 0.2349   
## GenderMale -0.2090 0.1919 -1.089 0.2761   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1182.9 on 899 degrees of freedom  
## Residual deviance: 1055.6 on 887 degrees of freedom  
## AIC: 1081.6  
##   
## Number of Fisher Scoring iterations: 4

Logistic Regression - Fit Smaller model based on significant variables from the regression using all variables. From the summary, we can say Loc\_score2, Perm\_score, Call\_score contribute significantly in the predictions.

glm.fit=glm(Obs\_Att\_class ~ Loc\_score2 + Perm\_score + Call\_score,  
 data=int\_df, family=binomial, subset=train)  
  
summary(glm.fit)

##   
## Call:  
## glm(formula = Obs\_Att\_class ~ Loc\_score2 + Perm\_score + Call\_score,   
## family = binomial, data = int\_df, subset = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0252 -1.0252 0.7287 0.8306 1.7982   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.3692 0.1660 -2.224 0.0261 \*   
## Loc\_score2 -0.3035 0.1574 -1.929 0.0538 .   
## Perm\_score 2.5858 0.3959 6.531 6.52e-11 \*\*\*  
## Call\_score -1.0262 0.4143 -2.477 0.0132 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1182.9 on 899 degrees of freedom  
## Residual deviance: 1062.6 on 896 degrees of freedom  
## AIC: 1070.6  
##   
## Number of Fisher Scoring iterations: 4

Loc\_score2, call score has negative coefficients

glm.probs=predict(glm.fit, newdata=int\_df[-train,],type="response")   
glm.pred=ifelse(glm.probs >0.5,1,0)  
table(glm.pred,Obs\_Att\_class[-train])

##   
## glm.pred 0 1  
## 0 48 37  
## 1 72 176

mean(glm.pred==Obs\_Att\_class[-train])

## [1] 0.6726727

Applying Logistic Regression Model, we would be able to predict 67.26 % correctly about the outcome of the interview.The Questions - “Have you obtained the necessary permission to start at the required time?”, “Has the call letter been shared?” attribute mainly to the predictions.

Decision Trees Model was created to check what are the important questions, which could predict the outcome.

attach(int\_df)

## The following objects are masked from int\_df (pos = 3):  
##   
## Alt\_no, Alt\_score, Alt\_tree, Call\_before\_3hrs, Call\_letter,  
## Call\_score, Call\_tree, Client.name, curr\_Location, Gender,  
## Job\_Location, Land\_score, Land\_tree, Landmark, Letter\_score,  
## Letter\_tree, Loc\_score1, Loc\_score2, Loc\_score3, Loc\_tree1,  
## Loc\_tree2, Loc\_tree3, Marital.Status, Native, Obs\_Att\_class,  
## Obs\_Att\_tree, Observed.Attendance, Perm\_score, Perm\_tree,  
## Permission, Rec\_Expectation, Res\_score, Res\_tree,  
## Resume\_prinout, Uns\_meet\_tree, Uns\_meeting\_score,  
## Unscheduled\_Meetings, Venue

## The following objects are masked from int\_df (pos = 5):  
##   
## Alt\_no, Call\_before\_3hrs, Call\_letter, Client.name,  
## curr\_Location, Gender, Job\_Location, Landmark, Marital.Status,  
## Native, Observed.Attendance, Permission, Rec\_Expectation,  
## Resume\_prinout, Unscheduled\_Meetings, Venue

require(tree)

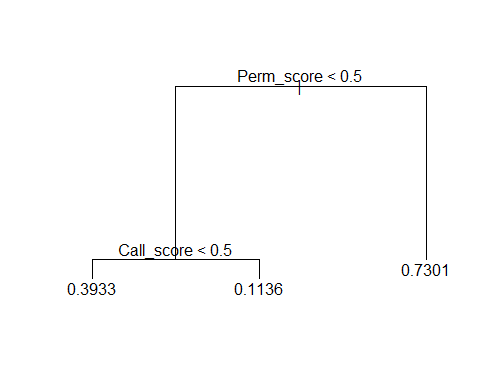
## Loading required package: tree

tree.intatt = tree(Obs\_Att\_class~ Loc\_score1 + Loc\_score2 + Loc\_score3 +   
 Perm\_score + Call\_score + Uns\_meeting\_score + Alt\_score +  
 Res\_score + Land\_score + Letter\_score +  
 Marital.Status + Gender, data=int\_df, subset = train)  
summary(tree.intatt)

##   
## Regression tree:  
## tree(formula = Obs\_Att\_class ~ Loc\_score1 + Loc\_score2 + Loc\_score3 +   
## Perm\_score + Call\_score + Uns\_meeting\_score + Alt\_score +   
## Res\_score + Land\_score + Letter\_score + Marital.Status +   
## Gender, data = int\_df, subset = train)  
## Variables actually used in tree construction:  
## [1] "Perm\_score" "Call\_score"  
## Number of terminal nodes: 3   
## Residual mean deviance: 0.2012 = 180.5 / 897   
## Distribution of residuals:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.7301 -0.3933 0.2699 0.0000 0.2699 0.8864

Regression tree has been created

plot(tree.intatt)  
text(tree.intatt, pretty=0)

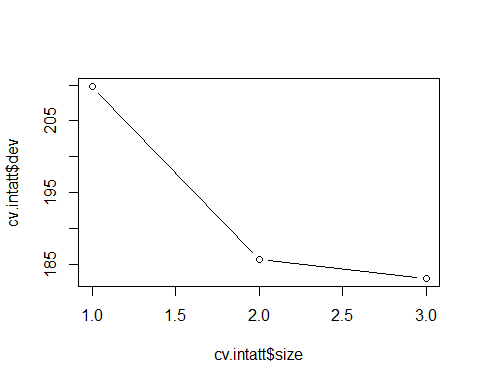


tree.intatt

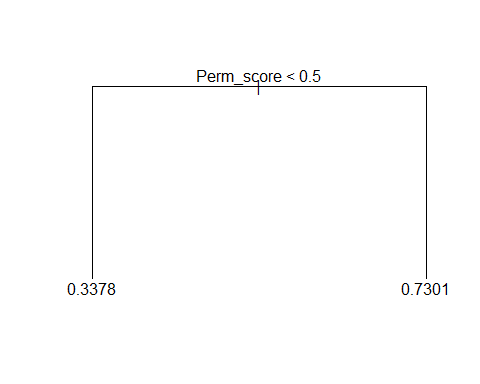
## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 900 209.000 0.6333   
## 2) Perm\_score < 0.5 222 49.660 0.3378   
## 4) Call\_score < 0.5 178 42.470 0.3933 \*  
## 5) Call\_score > 0.5 44 4.432 0.1136 \*  
## 3) Perm\_score > 0.5 678 133.600 0.7301 \*

Cross Validation

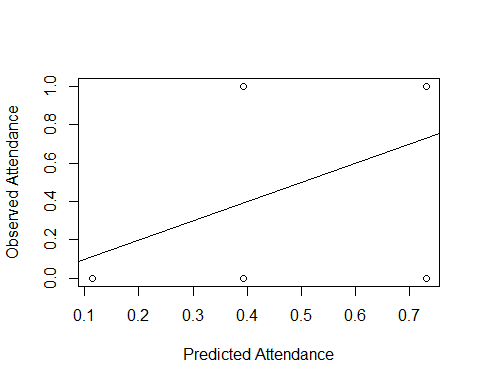
cv.intatt = cv.tree(tree.intatt)  
plot(cv.intatt$size ,cv.intatt$dev, type='b')

 Pruning

prune.intatt = prune.tree(tree.intatt, best = 2)  
plot(prune.intatt )  
text(prune.intatt, pretty= 0)



tree.pred = predict(tree.intatt, newdata=int\_df[-train, ])  
plot(tree.pred, Obs\_Att\_class[-train], xlab = "Predicted Attendance", ylab = "Observed Attendance")  
abline(0,1)



mean((tree.pred - Obs\_Att\_class[-train])^2)

## [1] 0.2114168

From applying Decision Trees model, we could say that based on the Question, “Have you obtained the necessary permission to start at the required time”, we would be able to predict 73.01% of the actual outcome.