# SATS 2019 - The Link Between Member Usage and Churn Prediction

# Amanda Limseth

setwd("~/Documents/BI /Master Thesis") # set working directory to "Master Thesis" folder (includes the SATSR.csv file)

list.files()

#INSTALL ALL PACKAGES NEEDED FOR THE PREDICTION

#install.packages("tidyverse")

#install.packages("ggthemes")

#install.packages("dplyr")

#install.packages("lubridate")

#install.packages("plyr")

#install.packages("Hmisc")

#install.packages("psych")

#install.packages("caret")

#install.packages("mice")

#install.packages("caTools")

#install.packages("ROCR")

#install.packages("stats")

#install.packages("car")

#install.packages("zoo")

#install.packages("tidyr")

#install.packages("pROC")

#install.packages("ggpubr")

#install.packages("AER")

#install.packages("stargazer")

#install.packages("lmtest")

#install.packages("mlbench")

#install.packages("MASS")

#install.packages("pROC")

#install.packages('visreg')

#install.packages('broom)

#install.packages("party")

#install.packages("rpart")

#install.packages("rpart.plot")

#install.packages("arm")

#install.packages("gbm")

#IMPORTING DATA PACKAGES

library(tidyverse)

library(ggthemes)

library(plyr)

library(dplyr)

library(lubridate)

library(ggplot2)

library(Hmisc)

library(psych)

library(caret)

library(forcats)

library(mice)

library(caTools)

library(ROCR)

library(stats)

library(car)

library(zoo)

library(tidyr)

library(pROC)

library(ggpubr)

library(AER)

library(stargazer)

library(lmtest)

library(MASS)

library(pROC)

library(visreg)

library(broom)

library(party)

library(rpart)

library(rpart.plot)

library(arm)

library(gbm)

### LOADING THE DATA ###

raw\_data <-read.table(file="SATSR.csv", sep=";", header=TRUE, stringsAsFactors = TRUE) # import data file

head(raw\_data) #returns the top 5 lines

names(raw\_data) #returns the names for each column headers

names(raw\_data)[1] <- "MemberID" #change the name of the MemberID column to MemberID

names(raw\_data)[21] <- "ActivityDate" #change the name of the “ActivityStartTime” column to “ActivityDate”

dim(raw\_data) #number of rows and variables

summary(raw\_data) #returns a summary of all variables in the data

######## DATA CLEANING AND EXPLORATION ########

#quick look at the data table and its variables

str(raw\_data)

glimpse(raw\_data)

#the data is stored in a data frame, but the types of variables need to be converted to the correct data class

#factors are variables that take on a limited number of values

#CONVERTING EACH VARIABLE TO THE BEST DATA CLASS

raw\_data$MemberID <- as.factor(raw\_data$MemberID)

raw\_data$HomeCenterID <- as.factor(raw\_data$HomeCenterID)

raw\_data$Age <- as.character(raw\_data$Age)

raw\_data$Age <- as.numeric(raw\_data$Age)

raw\_data$ActivityCenterID <- as.factor(raw\_data$ActivityCenterID)

raw\_data$ActivityNameID <- as.factor(raw\_data$ActivityNameID)

#REMOVE TIMESTAMPS FROM THE DATE CLASSES, AND CONVERT CLASSES

raw\_data$SubscriptionStartDate <- as.Date(raw\_data$SubscriptionStartDate, format="%d/%m/%Y") #convert to date

raw\_data$SubscriptionEndDate <- as.Date(raw\_data$SubscriptionEndDate, format="%d/%m/%Y")

raw\_data$ActivityDate <- as.Date(raw\_data$ActivityDate, format="%d/%m/%Y")

raw\_data$ActivityEndTime <- as.Date(raw\_data$ActivityEndTime, format="%d/%m/%Y")

raw\_data$SubscriptionStopDate <- as.Date(raw\_data$SubscriptionStopDate, format ="%d/%m/%Y")

raw\_data$SubscriptionBindingEndDate <- as.Date(raw\_data$SubscriptionBindingEndDate, format ="%d/%m/%Y")

glimpse(raw\_data)

#REMOVE DAY FROM DATE FOR AGGREGATING PER MONTH- CREATE NEW VARIABLE FOR MONTH\_YR

raw\_data$ActMonth\_Yr <- as.Date(raw\_data$ActivityDate, format = "%Y-%m")

raw\_data$date <- as.Date(raw\_data$ActMonth\_Yr, '%Y-%m')

#returning only Month-Year

raw\_data$ActMonth\_Yr <- substr(raw\_data$ActMonth\_Yr,0,7)

#CONVERTING MONTH\_YR TO ORDERED FACTOR, where the factor levels are ordered

raw\_data$ActMonth\_Yr <- as.ordered(raw\_data$ActMonth\_Yr)

#convert date to yearmon

raw\_data$date <- as.yearmon(raw\_data$date)

str(raw\_data)

#CHECKING THE LEVELS OF THE FACTORS

levels(raw\_data$City) #just 1 level (OSLO)

levels(raw\_data$Segment) #contains NULL-values

levels(raw\_data$SubscriptionName)

#some different ways of spelling ( All-Inclusive membership, All-Inclusive Membership, and All-Inclusive Membership no binding)

# Base EFT 12 months/maneder, no binding/uten binding

levels(raw\_data$MemberType)

levels(raw\_data$SubscriptionType) #contains Unknown-values

levels(raw\_data$ActivityName)

with(raw\_data, unique(SubscriptionName))

with(raw\_data, table(SubscriptionRenewalType))

 #majority of customers have recurring subscriptions

with(raw\_data, table(SubscriptionProductGroup))

levels(raw\_data$SubscriptionRenewalType)

#CORRECTING SOME MISSPELLINGS IN SUBSCRIPTIONNAMES

raw\_data[with(raw\_data, which(SubscriptionName == 'All-Inclusive membership')), 'SubscriptionName'] <- 'All-Inclusive Membership'

raw\_data[with(raw\_data, which(SubscriptionName == 'Base EFT 12 m????neder binding')), 'SubscriptionName'] <- 'Base EFT 12 months binding'

raw\_data[with(raw\_data, which(SubscriptionName == 'Base EFT uten binding')), 'SubscriptionName'] <- 'Base EFT no binding'

raw\_data[with(raw\_data, which(SubscriptionName == 'Base EFT maxprice no bindning')), 'SubscriptionName'] <- 'Base EFT no binding'

raw\_data[with(raw\_data, which(SubscriptionName == 'Gratis medlemskap')), 'SubscriptionName'] <- 'Free Trial TEMPORARY'

raw\_data[with(raw\_data, which(SubscriptionName == 'HiYoga EFT 12 months NB')), 'SubscriptionName'] <- 'HiYoga EFT 12 months'

with(raw\_data, table(SubscriptionName))

#CORRECTING MISSPELLINGS IN ACTIVITYNAMES

with(raw\_data, table(ActivityName))

raw\_data[with(raw\_data, which(ActivityName == 'Build Buildn Burn')), 'ActivityName'] <- 'Build n Burn'

raw\_data[with(raw\_data, which(ActivityName == 'Burn Buildn Burn')), 'ActivityName'] <- 'Build n Burn'

raw\_data[with(raw\_data, which(ActivityName == 'Buildn Burn')), 'ActivityName'] <- 'Build n Burn'

raw\_data[with(raw\_data, which(ActivityName == 'Crossing ')), 'ActivityName'] <- 'Crossing'

raw\_data[with(raw\_data, which(ActivityName == 'Energy Yoga ')), 'ActivityName'] <- 'Energy Yoga'

raw\_data[with(raw\_data, which(ActivityName == 'HIIT the Ground ')), 'ActivityName'] <- 'HIIT the Ground'

with(raw\_data, table(ActivityName))

#THERE ARE DIFFERENT VALUES IN THE VARIOUS COLUMNS WHEN DATA IS MISSING

#CONVERT NULL-VALUES, UNKNOWN AND NONE-VALUES TO NA'S

raw\_data[with(raw\_data, which(HomeCenterID == 'NULL')), "HomeCenterID"] <- NA

raw\_data[with(raw\_data, which(Age == "NULL")), "Age"] <- NA

raw\_data[with(raw\_data, which(PostalCode == 'NULL')), "PostalCode"] <- NA

raw\_data[with(raw\_data, which(City == 'NULL')), "City"] <- NA

raw\_data[with(raw\_data, which(Country == 'NULL')), "Country"] <- NA

raw\_data[with(raw\_data, which(Gender == 'NULL')), "Gender"] <- NA

raw\_data[with(raw\_data, which(PersonStatus == 'NULL')), "PersonStatus"] <- NA

raw\_data[with(raw\_data, which(Segment == 'NULL')), "Segment"] <- NA

raw\_data[with(raw\_data, which(MemberType == 'NULL')), "MemberType"] <- NA

raw\_data[with(raw\_data, which(SubscriptionRenewalType == 'NULL')), "SubscriptionRenewalType"] <- NA

raw\_data[with(raw\_data, which(SubscriptionState == 'NONE')), "SubscriptionState"] <- NA

raw\_data[with(raw\_data, which(SubscriptionSubState == 'NONE')), "SubscriptionSubState"] <- NA

raw\_data[with(raw\_data, which(SubscriptionName == 'NULL')), "SubscriptionName"] <- NA

raw\_data[with(raw\_data, which(SubscriptionProductGroup == 'NULL')), "SubscriptionProductGroup"] <- NA

raw\_data[with(raw\_data, which(SubscriptionType == 'Unknown')), "SubscriptionType"] <- NA

raw\_data[with(raw\_data, which(ActivityCenterID == 'NULL')), "ActivityCenterID"] <- NA

raw\_data[with(raw\_data, which(ActivityNameID == 'NULL')), "ActivityNameID"] <- NA

raw\_data[with(raw\_data, which(ActivityName == 'NULL')), "ActivityName"] <- NA

#AGE HAS MISSING VALUES, SO WILL REPLACE THE NA'S WITH THE MEAN AGE OF ALL OTHER MEMBERS

raw\_data$Age[is.na(raw\_data$Age)] <- median(raw\_data$Age, na.rm=TRUE)

#how many rows are not complete compared to those who are?

nrows <- nrow(raw\_data)

ncomplete <- sum(complete.cases(raw\_data))

ncomplete

#by removing the incomplete rows of data we would only have 0% of data left, because some of the rows have no data

#so we can not remove the incomplete rows

ncomplete/nrows

na\_info <- colSums(is.na(raw\_data))

view(na\_info)

#na\_info shows the distribution of missing values in the data set

#all SubscriptionSubState rows are NA's and majority of SubscriptionType is missing

#REMOVING INSUFFICIENT COLUMNS

raw\_data$SubscriptionSubState <- NULL

raw\_data$SubscriptionType <- NULL

raw\_data$SubscriptionStopDate <- NULL

#all customers belong to the same city and country, so there is no need to keep these variables

#REMOVING THE COLUMNS FOR CITY AND COUNTRY

raw\_data$City <- NULL

raw\_data$Country <- NULL

#the end time of activity, or ActivityNameID is not relevant for analysis

#REMOVE COLUMNS FOR ACTIVITYENDTIME AND ACTIVITYNAMEID

raw\_data$ActivityEndTime <- NULL

raw\_data$ActivityNameID <- NULL

######### GENERATE DUMMY (FACTOR) VARIABLES #########

#CONVERT GENDER TO DUMMY VARIABLE

raw\_data$GenderF <- 1

raw\_data[with(raw\_data, which(Gender == 'MALE')), 'GenderF'] <- 0

#CONVERT A DUMMY VARIABLE FOR CHURN (where is 1 when SubscriptionState == ENDED, and 0 == ACTIVE)

raw\_data$Churn <- 1

raw\_data[with(raw\_data, which(SubscriptionState == "ACTIVE")), "Churn"] <- 0

str(raw\_data)

#CONVERTING DUMMY VARIABLES FOR THE THREE MOST COMMON MEMBER TYPES

#STUDENT

raw\_data$Student <- 0

raw\_data[with(raw\_data, which(MemberType == "STUDENT")), "Student"] <- 1

#PRIVATE

raw\_data$Private <- 0

raw\_data[with(raw\_data, which(MemberType == "PRIVATE")), "Private"] <- 1

#CORPORATE

raw\_data$Corporate <- 0

raw\_data[with(raw\_data, which(MemberType == "CORPORATE")), "Corporate"] <- 1

#CONVERT DUMMY VARIABLES TO FACTORS

#Churn is the dependent variable for the model, which is what we are trying to predict

raw\_data$GenderF <- as.factor(raw\_data$GenderF)

raw\_data$Churn <- as.factor(raw\_data$Churn)

raw\_data$Student <- as.factor(raw\_data$Student)

raw\_data$Private <- as.factor(raw\_data$Private)

raw\_data$Corporate <- as.factor(raw\_data$Corporate)

#31,8% of the data contains members who have churned. So it's not considered a rare event, even though it's unbalanced

#and should be able to predict the Churn event in a model

prop.table(table(raw\_data$Churn))

prop.table((table(raw\_data$GenderF)))

#alter the gymactivity so that it only distinguishes between group classes (not each individual type of group class), and personal trainer, individual activityname

#MANUALLY DUMMY VARIABLES FOR THE THREE OPTIONS

#(gym, PT, and group class)

levels(raw\_data$ActivityName)

#DUMMY VARIABLE FOR GROUPCLASS

raw\_data$GroupClass <- 1

raw\_data[with(raw\_data, which(ActivityName == "Gym")), "GroupClass"] <- 0

raw\_data[with(raw\_data, which(ActivityName == "PT30")), "GroupClass"] <- 0

raw\_data[with(raw\_data, which(ActivityName == "PT60")), "GroupClass"] <- 0

raw\_data[with(raw\_data, which(ActivityName == "PTstart")), "GroupClass"] <- 0

raw\_data[with(raw\_data, which(ActivityName == "Ptstart1")), "GroupClass"] <- 0

raw\_data[with(raw\_data, which(ActivityName == "PTDuo60")), "GroupClass"] <- 0

#DUMMY VARIABLE FOR PERSONALTRAINER

raw\_data$PersonalTr <- 0

raw\_data[with(raw\_data, which(ActivityName == "PT30")), "PersonalTr"] <- 1

raw\_data[with(raw\_data, which(ActivityName == "PT60")), "PersonalTr"] <- 1

raw\_data[with(raw\_data, which(ActivityName == "PTstart")), "PersonalTr"] <- 1

raw\_data[with(raw\_data, which(ActivityName == "Ptstart1")), "PersonalTr"] <- 1

raw\_data[with(raw\_data, which(ActivityName == "PTDuo60")), "PersonalTr"] <- 1

#DUMMY VARIABLE FOR GYM

raw\_data$Gym <- 0

raw\_data[with(raw\_data, which(ActivityName == "Gym")), "Gym"] <- 1

#CONVERTING DUMMY VARIABLES TO FACTORS

raw\_data$GroupClass <- as.factor(raw\_data$GroupClass)

raw\_data$PersonalTr <- as.factor(raw\_data$PersonalTr)

raw\_data$Gym <- as.factor(raw\_data$Gym)

str(raw\_data)

#there are several variations of names for the subscriptions, but any contract that is not binding has monthly renewal intervals

#CREATE DUMMY VARIABLES FOR YEARLY AND MONTHLY-RENEWAL

levels(raw\_data$SubscriptionName)

raw\_data$MonthlyRenewal <- 0

raw\_data[with(raw\_data, which (SubscriptionName == "All-Inclusive Membership no binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Badeland BARN EFT no binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Badeland EFT no binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Base EFT maxprice no binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Base EFT no binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Base EFT uten binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Base Together EFT no binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Nordic, Daytime, Gym training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Nordic, Group training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Nordic, Gym training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "One club, Daytime, Gym training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "One club, Group training and concepts, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "One club, Group training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "One club, Gym training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Region, Daytime, Group training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Region, Group training and concepts, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Region, Group training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Region, Gym training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Two clubs, Daytime, Gym training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Two clubs, Group training and concepts, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Two clubs, Group training, No binding")), "MonthlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Two clubs, Gym training, No binding")), "MonthlyRenewal"] <- 1

raw\_data$YearlyRenewal <- 0

raw\_data[with(raw\_data, which (SubscriptionName == "All-Inclusive membership")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "All-Inclusive Membership")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Badeland BARN EFT 12 months")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Badeland EFT 12 months")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Base Drammen EFT exkl GX")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Base EFT 12 mÃ¥neder binding")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Base EFT 12 months binding")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Base Together EFT 12 months")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Corporate City Cash 12 months")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Corporate City EFT 12 months")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Corporate Priority EFT 12 months")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Double EFT 12 months")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Fitness Only EFT 12 months")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "HiYoga EFT 12 months")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "HiYoga EFT 12 months NB")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Nordic, Group training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Nordic, Group training and concepts")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Nordic, Gym training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "One club, Daytime, Group training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "One club, Daytime, Gym training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "One club, Group training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "One club, Group training and concepts")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "One club, Gym training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Priority EFT 12 months")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Region, Daytime, Group training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Region, Daytime, Gym training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Region, Group training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Region, Group training and concepts")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Region, Gym training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Two clubs, Daytime, Group training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Two clubs, Daytime, Gym training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Two clubs, Group training")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Two clubs, Group training and concepts")), "YearlyRenewal"] <- 1

raw\_data[with(raw\_data, which (SubscriptionName == "Two clubs, Gym training")), "YearlyRenewal"] <- 1

#CONVERTING DUMMY VARIABLES TO FACTORS

raw\_data$MonthlyRenewal <- as.factor(raw\_data$MonthlyRenewal)

raw\_data$YearlyRenewal <- as.factor(raw\_data$YearlyRenewal)

str(raw\_data)

###### AGGREGATE MONTHLY ACTIVITY LEVELS PER UNIQUE ID ######

# TOTAL ROWS PER UNIQUE ID - TOTAL OF ALL ACTIVITIES COMBINED

total.aggregated <- count(raw\_data, c("MemberID"))

#WANT TO DISPLAY THE PERCENTAGE USAGE OF EACH ACTIVITY TYPE PER MEMBER

#TOTAL ROWS PER ACTIVITY TYPE, PER UNIQUE ID

totals.aggregated <- count(raw\_data, c("MemberID", "GroupClass", "PersonalTr", "Gym"))

totals.aggregated$Age <- raw\_data$Age[match(totals.aggregated$MemberID, raw\_data$MemberID)]

#MERGE INTO ONE TABLE WITH ALL PROPORTIONS OF ACTIVITIES

all.activities <- merge(total.aggregated, totals.aggregated, by.x=c("MemberID"), by.y=c("MemberID"), all=TRUE)

#renaming the columns

names(all.activities)[2] <- "TotalFreq"

names(all.activities)[6] <- "ActivityFreq"

with(raw\_data, plot(Age))

#DIVIDE TOTAL NUMBER OF PERSONALTR AND GROUPCLASS-ACTIVITY WITH THE TOTAL FREQUENCY PER MEMBER

#WANT THE AVERAGE USE OF THE DIFFERENT ACTIVITIES

all.activities$Proportion <- all.activities$ActivityFreq / all.activities$TotalFreq

#insert new columns instead of the dummy variables for the type of activitiy (With proportion instead)

#creating column for groupclass proportion

all.activities$GroupClass <- as.numeric(all.activities$GroupClass)

all.activities$GroupClass[all.activities$GroupClass==1] <- 0

all.activities$GroupClass[all.activities$GroupClass==2] <- 1

all.activities$GroupProp <- all.activities$GroupClass \* all.activities$Proportion

#creating column for personaltr proportion

all.activities$PersonalTr <- as.numeric(all.activities$PersonalTr)

all.activities$PersonalTr[all.activities$PersonalTr==1] <- 0

all.activities$PersonalTr[all.activities$PersonalTr==2] <- 1

all.activities$PTProp <- all.activities$PersonalTr \* all.activities$Proportion

#creating column for gym proportion

all.activities$Gym <- as.numeric(all.activities$Gym)

all.activities$Gym[all.activities$Gym==1] <- 0

all.activities$Gym[all.activities$Gym==2] <- 1

all.activities$GymProp <- all.activities$Gym \* all.activities$Proportion

#REMOVING PROPORTION COLUMN

all.activities$Proportion <- NULL

#FILL IN EMPTY ROWS WITH SAME VALUES FOR EACH MEMBER

#FOR GROUP CLASS

all.activities$GroupProp[all.activities$GroupProp == 0] <- NA

#FILLING IN PROPORTIONS FOR GROUP CLASS WHERE THERE IS NA'S

all.activities <- all.activities %>%

 group\_by(MemberID) %>%

 fill(GroupProp, .direction = "up")

all.activities <- all.activities %>%

 group\_by(MemberID) %>%

 fill(GroupProp, .direction = "down")

#RETURNING NA'S TO 0 PROPORTION

all.activities$GroupProp[is.na(all.activities$GroupProp)] <- 0

#FOR PERSONAL TRAINER

all.activities$PTProp[all.activities$PTProp == 0] <- NA

#FILLING IN PROPORTIONS FOR PERSONAL TRAINER WHERE THERE IS NA'S

all.activities <- all.activities %>%

 group\_by(MemberID) %>%

 fill(PTProp, .direction = "up")

all.activities <- all.activities %>%

 group\_by(MemberID) %>%

 fill(PTProp, .direction = "down")

#RETURNING NA'S TO 0 PROPORTION

all.activities$PTProp[is.na(all.activities$PTProp)] <- 0

#FOR GYM

all.activities$GymProp[all.activities$GymProp == 0] <- NA

#FILLING IN PROPORTIONS FOR GYM WHERE THERE IS NA'S

all.activities <- all.activities %>%

 group\_by(MemberID) %>%

 fill(GymProp, .direction = "up")

all.activities <- all.activities %>%

 group\_by(MemberID) %>%

 fill(GymProp, .direction = "down")

#RETURNING NA'S TO 0 PROPORTION

all.activities$GymProp[is.na(all.activities$GymProp)] <- 0

#REMOVING DUMMY VARIABLES FROM THE TABLE

all.activities$GroupClass <- NULL

all.activities$PersonalTr <- NULL

all.activities$Gym <- NULL

all.activities$ActivityFreq <- NULL

#CREATE A GRAPH TO SHOW THE DISTRIBUTION OF ACTIVITY TYPE FOR THE MEMBERS

with(totals.aggregated, plot(x = PersonalTr, y = freq, main = 'Frequency of PersonalTrainer'))

with(totals.aggregated, plot(x = Gym, y = freq, main = 'Frequency of Gym'))

with(totals.aggregated, plot(x = GroupClass, y = freq, main = 'Frequency of Groupclass'))

#lowest total is within personaltrainer (at around 300, but majority much lower)

#while both groupclass and gym see outliers up to the 700 mark, while the 3rd quartile (75% of the customers) fall below 80

#there exists many large outliers in all categories, but they don’t seem like they are that unnaturally high that they necessarily are mistakes

#should consider whether they should stay in the data set, or if some outliers will be removed

range(totals.aggregated$freq) #min 1, max 712

#CHECKING DISTRIBUTION OF CONTRACT TYPE (MONTHLY, AND YEARLY)

prop.table((table(raw\_data$MonthlyRenewal)))

prop.table((table(raw\_data$YearlyRenewal)))

#9,2% of the members have monthly renewal contracts, while 89,4% have yearly contracts

#not enough data to model for monthly, will focus on yearly

#MONTHLY AGGREGATED ACTIVITY PER UNIQUE ID

month.aggregated <- count(raw\_data, c("MemberID", "ActMonth\_Yr", "Churn", "date", "YearlyRenewal"))

#WITH ADDITIONAL VARIABLES

month.aggregated <- count(raw\_data, c("MemberID", "GenderF", "ActMonth\_Yr", "Churn", "YearlyRenewal"))

#with(month.aggregated, plot(MemberID, freq, main = "Total Frequency of Activity per MemberID"))

#with(month.aggregated, plot(x = ActMonth\_Yr, y = freq, main = "Monthly Aggregated Activity"))

#CALCULATING TOTAL ACTIVE MONTHS PER MEMBER ID

#in order to include the months where the member has no visits to the center

month.aggregated$n\_months <- 12 \* as.numeric(substr(month.aggregated$ActMonth\_Yr, 1, 4)) + as.numeric(substr(month.aggregated$ActMonth\_Yr, 6, 7))

#what month in the sequence of levels the customer is active

month.aggregated$month\_count <- month.aggregated$n\_months - min(month.aggregated$n\_months)

month.aggregated$n\_months <- NULL

#CREATE NEW VARIABLE FOR TENURE - months active (= tenure)

#for each month, what is the contract duration(tenure) at that point

months.aggregated <- lapply(split(month.aggregated, month.aggregated$MemberID), function(x) {

 x$TenureMonths <- x$month\_count - x$month\_count[1]

 x})

months.aggregated <- do.call(rbind, months.aggregated)

row.names(months.aggregated) <- NULL

months.aggregated$TenureMonths <- as.integer(months.aggregated$TenureMonths)

#removing month\_count column from table

months.aggregated$month\_count <- NULL

view(months.aggregated)

#CREATE NEW VARIABLE FOR TENURE IN RAW\_DATA

#(will return how many weeks the customer has been under contract at the time of visit)

raw\_data$TenureWeeks <- difftime(as.Date(raw\_data$ActivityDate), as.Date(raw\_data$SubscriptionStartDate), units = "weeks")

range(raw\_data$TenureWeeks) #min 0, max 125

mean(raw\_data$TenureWeeks)

raw\_data$TenureMonths <- raw\_data$TenureWeeks/4

range(raw\_data$TenureMonths) #min 0, max 31,3

mean(raw\_data$TenureMonths)

### INSERT MONTHS WHERE THE ACTIVITY IS AT 0

#finds the gaps in the sequence of the tenure months over the time period

new\_data <- months.aggregated %>%

 group\_by(MemberID) %>%

 tidyr::complete(TenureMonths = full\_seq(TenureMonths, 1), fill = list(freq = 0, Churn =0))

view(new\_data)

#FILLING IN NA-ROWS IN YEARLYRENEWAL AND GENDERF WITH PREVIOUS VALUE

new\_data <- fill(new\_data, YearlyRenewal, .direction="down")

new\_data <- fill(new\_data, GenderF, .direction="down")

#MANIPULATE SO THAT IF THE MEMBER CHURNS, ONLY THE LAST MONTH IS A "1", AND THE OTHERS REMAIN "0"'S

#this will highlight what month is the last active one for each member

#extracting the max tenure value for each member (the last known active month)

maxsub <- new\_data %>%

 group\_by(MemberID) %>%

 filter(row\_number(TenureMonths)==n())

#converting all Churn variables to 0, so can later combine with the values found in maxsub

all0 <- new\_data %>%

 group\_by(MemberID) %>%

 mutate(Churn = ifelse(Churn == 1, 0, 0))

#REPLACE CURRENT VALUE IN ALL0, WITH THE VALUES IN MAXSUB

#will put 1, in the last column for the members who have churned

newchurn <- merge(all0, maxsub, by.x=c("MemberID", "TenureMonths"), by.y=c("MemberID", "TenureMonths"), all=TRUE)

#insert Churn.y column from new churn into new\_data

new\_data$newChurn <- newchurn$Churn.y

#REPLACE NA'S WITH 0, FOR A COMPLETE CHURN COLUMN

new\_data$newChurn[is.na(new\_data$newChurn)] <- 0

#remove old Churn column from data set

new\_data$Churn <- NULL

with(new\_data, plot(x = ActMonth\_Yr, y = freq, main = "Monthly Aggregated Activity"))

#new\_data$ActMonth\_Yr <- NULL

#CREATING NEW VARIABLE WITH MONTHLY AVERAGE FOR EACH CUSTOMER

#average monthly is sum(freq) divided by (max(month\_count)-min(month\_count))

new\_data$MonthlyAverage <- ave(new\_data$freq, new\_data$MemberID)

view(new\_data)

#inspecting the range, and mean of the new variables

range(new\_data$freq) #min 0, max 61

mean(new\_data$freq) #6.21

range(new\_data$MonthlyAverage) #min 0.182, max 32.92

mean(new\_data$MonthlyAverage) #6.21

#quantile returns quantiles corresponding to the given probabilities

quantile(new\_data$freq)

# 0% = 0, 25% = 2, 50% = 5, 75% = 9, 100% = 61

# up to the 75th quantile is below 10, while the jump to the 100th is very big

# 75th percentile means that 75% of all the customers frequency falls below 10,

# while all customers fall below 61

quantile(new\_data$MonthlyAverage)

# 0% = 0.182, 25% = 3.22, 50% = 5.33, 75% = 8.19, 100% = 32,92

#group tenure based on length for each member

min(new\_data$TenureMonths) #0

max(new\_data$TenureMonths) #28, we have a total of 29 months in the data, but don't know the previous history

#PLOTTING DISTRIBUTION OF RENEWAL TYPES

#they are the same for all the members (used to create the subset)

ggplot(renewal\_dec) + geom\_bar(mapping = aes (x = YearlyRenewal))

#SEPARATE OUT THE MEMBERS WHO DON'T HAVE YEARLY CONTRACT RENEWALS

#because the percentage of monthly renewal present in the data is pretty low, the model will focus only on the yearly

##NEW TABLE WITH YEARLY ROWS, AT THE END OF CONTRACT PERIOD - DOES THE MEMBER CHURN OR STAY

#only the members who are in the data set beyond 12 months have made the renewal decision - these are the only relevant rows for prediction

renewal\_dec <- subset(new\_data, TenureMonths == 12,

 select=c(MemberID, TenureMonths, newChurn, GenderF, YearlyRenewal, MonthlyAverage))

#NEW SUBSET WITH ONLY YEARLY RENEWALS

renew <- subset(renewal\_dec, YearlyRenewal == 1,

 select=c(MemberID, TenureMonths, newChurn, GenderF, YearlyRenewal, MonthlyAverage))

## INCLUDE MORE USAGE INFORMATION: INSERTING TOTAL FREQUENCY OF ACTIVITY

renew$TotalFreq <- all.activities$TotalFreq[match(renew$MemberID, all.activities$MemberID)]

#update monthly average

renew$MonthlyAverage1 <- renew$TotalFreq / 12

#REMOVING THE OLD VARIABLE, AND UPDATING THE NAME FOR THE NEW ONE

renew$MonthlyAverage <- NULL

names(renew)[7] <- "MonthlyAverage"

#INCLUDE ACTIVITY AS PROPORTIONS FROM ALL.ACTIVITIES TABLE TO REPRESENT THE DISTRIBUTION OF USAGE

#THIS SHOULD HELP ACCOUNT FOR THE FACT THAT THE MEMBERS HAVEN'T BEEN ACTIVE FOR THE SAME LENGTH OF PERIOD

#INSERTING AGE COLUMN

renew$Age <- raw\_data$Age[match(renew$MemberID, raw\_data$MemberID)]

#MERGE TO INCLUDE THE PROPORTION OF THE ACTIVITY TYPES WITH THE DUMMY VARIABLES

renew$GroupProp <- (all.activities$GroupProp[match(renew$MemberID, all.activities$MemberID)]) \* 100

renew$PTProp <- (all.activities$PTProp[match(renew$MemberID, all.activities$MemberID)]) \* 100

renew$GymProp <- (all.activities$GymProp[match(renew$MemberID, all.activities$MemberID)]) \* 100

#MERGE WITH RAW\_DATA TO INCLUDE MEMBER TYPES IN THE FINAL DATA

renew$Student <- raw\_data$Student[match(renew$MemberID, raw\_data$MemberID)]

renew$Private <- raw\_data$Private[match(renew$MemberID, raw\_data$MemberID)]

renew$Corporate <- raw\_data$Corporate[match(renew$MemberID, raw\_data$MemberID)]

#INCLUDE LAST USAGE COLUMNS - TO SHOW THE AGGREGATED ACTIVITY IN THE 12TH, 11TH AND 10TH MONTH FOR EACH MEMBER

#creating individual subsets for each of the months, based on the tenure variable where freq is recorded

months.agg12 <- subset(new\_data, TenureMonths == 12,

 select =c(MemberID, TenureMonths,freq))

months.agg11 <- subset(new\_data, TenureMonths == 11,

 select =c(MemberID, TenureMonths,freq))

months.agg10 <- subset(new\_data, TenureMonths == 10,

 select =c(MemberID, TenureMonths,freq))

months.agg9 <- subset(new\_data, TenureMonths == 9,

 select =c(MemberID, TenureMonths,freq))

months.agg8 <- subset(new\_data, TenureMonths == 8,

 select =c(MemberID, TenureMonths,freq))

#INSERTING THE FREQ FROM THE MONTH.AGG'S INTO THE RENEW-DATA

renew$ActMonth12 <- months.agg12$freq[match(renew$MemberID, months.agg12$MemberID)]

renew$ActMonth11 <- months.agg11$freq[match(renew$MemberID, months.agg11$MemberID)]

renew$ActMonth10 <- months.agg10$freq[match(renew$MemberID, months.agg10$MemberID)]

renew$ActMonth9 <- months.agg9$freq[match(renew$MemberID, months.agg9$MemberID)]

renew$ActMonth8 <- months.agg8$freq[match(renew$MemberID, months.agg8$MemberID)]

#CALCULATING THE DIFFERENCE IN MONTH 10,11, AND 12 FROM THE AVERAGE

#multiplied by 100 to give the percentage in relation to monthly average (100 % is same level, while under means the months i lower than the average and vice versa)

renew$Month12Prop <- (renew$ActMonth12/renew$MonthlyAverage) \* 100

renew$Month11Prop <- (renew$ActMonth11/renew$MonthlyAverage) \* 100

renew$Month10Prop <- (renew$ActMonth10/renew$MonthlyAverage) \* 100

renew$Month9Prop <- (renew$ActMonth9/renew$MonthlyAverage) \* 100

renew$Month8Prop <- (renew$ActMonth8/renew$MonthlyAverage) \* 100

#REMOVE TENUREMOTNHS AND YEARLYRENEWAL FROM THE DATA

renew$TenureMonths <- NULL

renew$YearlyRenewal <- NULL

renew$ActMonth12 <- NULL

renew$ActMonth11 <- NULL

renew$ActMonth10 <- NULL

renew$ActMonth9 <- NULL

renew$ActMonth8 <- NULL

renew$MemberID <- NULL

renew$TotalFreq <- NULL

renew$Month9Prop <- NULL

renew$Month8Prop <- NULL

#FINAL DATA SET FOR MODEL BUILDING

names(renew)[1] <- "Churn"

summary(renew)

renew <- as.data.frame(renew)

str(renew)

########SUMMARY OF ALL DATA SET VARIABLES###########

######### CONTINUED EXPLANATORY ANALYSIS ################

#PLOTTING TO SEE DISTRIBUTION OF THE DIFFERENT VARIABLES

#investigate before continuing with building regression models

#but all ID's are combined, need to extract unique ID's to get correct distribution

ggplot(raw\_data) + geom\_bar(mapping = aes(x=MemberType))

# explore the distribution of members

#consider removing friend, senior, and staff segment from the data? Not many in either type

ggplot(raw\_data) + geom\_bar(mapping = aes (x=Segment))

ggplot(raw\_data) + geom\_bar(mapping = aes (x=SubscriptionRenewalType))

ggplot(raw\_data) + geom\_bar(mapping = aes (x=PersonStatus))

ggplot(renew) + geom\_bar(mapping = aes (x = GenderF))

ggplot(renew) + geom\_bar(mapping = aes (x = Student))

ggplot(renew) + geom\_bar(mapping = aes (x = Private))

ggplot(renew) + geom\_bar(mapping = aes (x = Corporate))

ggplot(new\_data) + geom\_bar(mapping = aes (x = TenureMonths))

ggplot(new\_data) + geom\_bar(mapping = aes (x = tenure\_group))

ggplot(new\_data) + geom\_bar(mapping = aes (x = freq))

ggplot(new\_data) + geom\_bar(mapping = aes (x = MonthlyAverage))

ggplot(renew) + geom\_bar(mapping = aes (x = MonthlyAverage))

ggplot(maxsub) + geom\_bar(mapping = aes (x = Churn))

 # half of the members in the raw data has churned

ggplot(renew) + geom\_bar(mapping = aes (x = Churn)) # after 12 months, the spread of churn has gone considerably down

#imbalance problem is much more considerable

#HISTOGRAM OF AGE

with(raw\_data, hist(Age, col = 'grey'))

#HISTOGRAM OF AGE WITH SEGMENTS

ggplot(raw\_data, aes(Age)) + geom\_histogram(aes(fill=Segment), color='black', binwidth = 5)

#shows that the majority within all age groups are from enthusiasts, or passives

#HISTOGRAM OF AGE WITH GENDER

ggplot(raw\_data, aes(Age)) + geom\_histogram(aes(fill=Gender), color='black', binwidth = 5)

#HISTOGRAM OF AGE WITH MEMBERTYPES

ggplot(raw\_data, aes(Age)) + geom\_histogram(aes(fill=MemberType), color = 'black', binwidth = 2)

#HISTOGRAM OF TENURE MONTHS

with(new\_data, hist(TenureMonths, col = 'grey'))

with(new\_data, hist(TenureMonths, col = 'grey'))

#tenure months shows a right skew (tail on the right side), where the majority of the customers are below 10 months

#this is highly dependent on the length of the data set, where we only have access to data from 2017-2019, and don't know their history before this time period

with(renew, hist(TenureMonths, col = 'grey'))

# in the renew data, there is only either 12 or 24 months

#these are the months where the members have to decide if they sign out of their membership

ggplot(raw\_data, aes(TenureMonths)) + geom\_histogram(aes(fill=Segment), color='black', binwidth = 5)

ggplot(raw\_data, aes(TenureMonths)) + geom\_histogram(aes(fill=Gender), color='black', binwidth = 5)

#doesn't seem to be any difference between gender and tenure months. Even distribution in all bins

raw\_data$logTenureMonths <- with(raw\_data, log(TenureMonths))

with(raw\_data, hist(logTenureMonths, col = 'grey'))

#log distribution is more evenly distributed, but now somewhat leftskewed

ggplot(raw\_data, aes(logTenureMonths)) + geom\_histogram(aes(fill=Segment), color='black', binwidth = 1)

#HISTOGRAM OF FREQUENCY OF MONTHLY ACTIVITY

with(month.aggregated, hist(freq, col = 'grey'))

with(renew, hist(MonthlyAverage, col = "grey"))

with(renew, hist(Age, col = "grey"))

#the proportions are unevenly distributed

#may need transformations

with(renew, hist(GroupProp, col = "grey")) #all proportions are highly skewed

with(renew, hist(PTProp, col = "grey")) #so they can be analyzed in buckets to show their effect more clearly

with(renew, hist(GymProp, col = "grey"))

#histogram of average monthly activity

with(new\_data, hist(MonthlyAverage, col='grey', main = "Average Monthly Activity"))

with(renew1, hist(MonthlyAverage, col='orange')) #slightly skewed

with(raw\_data, plot(Age, TenureMonths))

#shows that there seems to be no relationship between age and tenure months, the spread is even between

#all ages between 15-80

#there are some outliers with different behavior in the most extreme age brackets

#PLOT OF AVERAGE MONTHLY ACTIVITY OVER MONTHS

with(new\_data, plot(ActMonth\_Yr, MonthlyAverage, main = "Average Monthly Activity"))

with(new\_data, plot(ActMonth\_Yr, TenureMonths, main = "Spread of Tenure over the Collection Period"))

#shows that there are newly signed customers in every recorded month

#there will be some customers that will not have the same time period collected because of the short length

#the longer the collection period, the wider the gap between the difference in tenure between the members becomes

#fast description of all variables in dataset. 10 000 distinct customers, over 79 distinct centers

#about even distribution between genders, more females (53/47)

#Enthusiasts - high frequency (39% of customers belong to segment, while 33,7% are passives)

#only 0,4% have a prepaid subscription (majority are recurring)

new\_data %>% Hmisc::describe()

renew %>% Hmisc::describe()

activityPerType <- count(raw\_data$Gym, raw\_data$PersonalTr, raw\_data$GroupClass)

view(activityPerType)

#one type of activity is outlier (most popular type of activity = gym)

#limiting to showing top 10 most frequent, instead of all 200 types of activities

with(activityPerType, plot(x, freq, main = "Total Frequency per Type of Activity"))

#check distribution between segments and gender

with(raw\_data, plot(Segment, Gender, main= "Gender and Segment Distribution"))

with(raw\_data, plot(MemberType, Gender))

str(raw\_data)

#distribution of churn in the various variables

ggplot(renew, aes(MonthlyAverage)) + geom\_histogram(aes(fill = Churn), color = 'black', binwidth = 2)

ggplot(renew, aes(Age)) + geom\_histogram(aes(fill = Churn), color = 'black', binwidth = 2)

#shows that churn isn't limited to having a low or null monthly average. the majority is placed within the 5-bucket

ggplot(renew, aes(Month12Prop)) + geom\_histogram(aes(fill = Churn), color = 'black', binwidth = 2)

#in last months activity, the majority of churn is however as expected, with no activity

ggplot(renew, aes(TotalFreq)) + geom\_histogram(aes(fill = Churn), color = 'black', binwidth = 20)

#the members who have the highest total frequency, see little churn

str(renew1)

#CHECK FOR CORRELATED VARIABLES BEFORE MOVING ON TO MODELING

cor.test(renew1$MonthlyAverage, renew1$TotalFreq) #correlation = 0.904

#these two predictors are highly correlated, and therefore only one should be included in the model (not both)

cor.test(renew$GroupProp, renew$GymProp) #correlation = -0.96, strong negative relationship

cor.test(renew$GymProp, renew$PTProp) #correlation = -0.16

cor.test(renew$GroupProp, renew$PTProp) #correlation =-0.11

cor.test(renew$MonthlyAverage, renew$GymProp) #correlation = 0.092, virtually no association between the two

cor.test(renew$MonthlyAverage, renew$Month12Prop) #correlation = - 0.06

cor.test(renew1$MonthlyAverage, renew1$Age) #correlation = -0.081

#PLOT TO SHOW THE CORRELATION BETWEEN MONTHLYAVG AND TOTALFREQ

ggscatter(renew1, x = "MonthlyAverage", y = "TotalFreq",

 add = "reg.line", conf.int = TRUE,

 cor.coef = TRUE, cor.method = "pearson")

#PLOT OF CORRELATION BETWEEN MONTHLY AVERAGE AND GYMPROP

ggscatter(renew, x = "MonthlyAverage", y = "GymProp",

 cor.coef = TRUE, cor.method = "pearson")

ggscatter(renew1, x = "MonthlyAverage", y = "Gym\_bucket",

 cor.coef = TRUE, cor.method = "pearson")

#PLOT OF CORRELATION BETWEEN MONTHLY AVERAGE AND PTPROP

ggscatter(renew, x = "MonthlyAverage", y = "PTProp",

 cor.coef = TRUE, cor.method = "pearson")

#correlation coefficient = 0.0093, virtually no relationship between the two variables

ggscatter(renew1, x = "MonthlyAverage", y = "PT\_bucket",

 cor.coef = TRUE, cor.method = "pearson")

#PLOT OF CORRELATION BETWEEN MONTHLY AVERAGE AND LASTMONTHACT

ggscatter(renew, x = "MonthlyAverage", y = "LastMonthAct",

 cor.coef = TRUE, cor.method = "pearson")

#somewhat correlated, but some variability, so could be interesting to include in model

#PLOT OF CORRELATION BETWEEN GROUPPROP AND GYMPROP

ggscatter(renew, x = "GroupProp", y = "GymProp",

 cor.coef = TRUE, cor.method = "pearson")

#PLOT OF CORRELATION BETWEEN GYMPROP AND PTPROP

ggscatter(renew, x = "GymProp", y = "PTProp",

 cor.coef = TRUE, cor.method = "pearson")

#PLOT OF CORRELATION BETWEEN GROUPPROP AND PTPROP

ggscatter(renew, x = "GroupProp", y = "PTProp",

 cor.coef = TRUE, cor.method = "pearson")

#EXPLORING THE DEPENDENT AND INDEPENDENT RELATIONSHIPS

plot(Churn ~ MonthlyAverage, renew)

#the plot shows that the higher activity level, the less likely to churn, but with some exceptions

plot(Churn ~ Month12Propt, renew)

#the plot shows that the higher the activity level in the last month is, the less likely the members are to churn

#but even though the proportion of churn is higher in the lower activity levels, churn still occurs in highly active members

#BOXPLOT OF CHURN AND MONTHLYAVERAGE

boxplot(MonthlyAverage ~ Churn,

 xlab = "Churn (1/0)", ylab = "MonthlyAverage", main = "Churn Distribution with Monthly Average", renew)

boxplot(Month12Prop ~ Churn,

 xlab = "Churn (1/0)", ylab = "Month12Prop", main = "Churn Distribution with 12th Month Proportion", renew)

#shows that the last monthly average is lower for those members who end up churning

boxplot(Age ~ Churn,

 xlab = "Churn (1/0)", ylab = "Age", main = "Churn Distribution with Age", renew)

#average age is similar between the groups who churn and don't churn

boxplot(GroupProp ~ Churn,

 xlab = "Churn (1/0)", ylab = "Group Proportion", main = "Churn Distribution with Group Proportion", renew)

boxplot(GymProp ~ Churn,

 xlab = "Churn (1/0)", ylab = "Gym Proportion", main = "Churn Distribution with Gym Proportion", renew)

boxplot(PTProp ~ Churn,

 xlab = "Churn (1/0)", ylab = "PT Proportion", main = "Churn Distribution with PT Proportion", renew)

summary(renew)

############# MODEL BUILDING ############

#TRAINING THE MODEL BY SPLITTING UP THE DATA INTO TRAIN AND TEST SETS

set.seed(90)

split <- sample.split(renew$Churn, SplitRatio = 0.7)

train <- subset(renew, split == TRUE)

test <- subset(renew, split == FALSE)

paste("train sample size: ", dim(train)[1]) #2937

paste("test sample size: ", dim(test)[1]) #1259

#classification for prediction requires a binary model

prop.table(table(train$Churn))

#93.8 % non-churn, and 6.2 % churn cases present in the train set

#THERE IS A CLASS IMBALANCE PROBLEM

#ADJUST CUT-OFF VALUE TO THE OCCURRENCE OF CHURN IN THE DATA (at 6 %)

#baseline model is using the imbalance against itself, and predicting all members as non churning

#will return an accuracy score of 94%, but no churn cases are identified

#comparable models will be binary logit, probit, and classification tree model

#LOGIT MODEL, glm = general linear model

#prediction is binomial as the Churn variable is a binary variable, the customer either churns or it doesn't

#full simple logistic model (with all explanatory variables added + interaction terms)

logit1 <- glm(Churn ~ . + MonthlyAverage \* GroupProp + MonthlyAverage \* PTProp, family = binomial(link = 'logit'), data = train)

#link = 'logit', returns the log odds

summary(logit1)

broom::glance(logit1)

#plot the model

plot(logit1)

#AIC stepwise removal of insignificant variables

logit\_2 <- stepAIC(logit1) # plus manual removal of insignificant variables

#REDUCED LOGIT MODEL

#AIC = 1238

logit2 <- glm(Churn ~ MonthlyAverage + Age + GroupProp + Private +

 Month11Prop + Month10Prop + MonthlyAverage:GroupProp, family = binomial(link = "logit"), data = train)

#month 12 is not significant, maybe it is already too late to make a difference

#so the prediction should be focused on looking at the activity in month 10 and 11 to predict ahead of time

summary(logit2)

broom::augment(logit2)

#ANOVa table shows the analysis of the table of deviance

anova(logit2, test ="Chisq")

#the difference between the NULL model shows the model is doing better than the model with only intercept

#the deviance drops for each variable added (first variable with 100, and additional 10 for next two variables)

#the low p-value for each variable added means that the model can explain more with the variable added

#install.packages("margins")

library(margins)

#THE MARGINAL EFFECT : LOGIT

#LOGIT MODEL AVERAGE MARGINAL EFFECTS

#dlogis for the logistic distribution

LogitScalar <- mean(dlogis(predict(logit2, type = "link")))

LogitScalar \* coef(logit2)

#SHOWING THE INTERACTION EFFECTS OF THE VARIABLES

#install.packages(“intercations”)

library(interactions)

interactions::interact\_plot(logit2, pred = Age, modx = MonthlyAverage, interval = TRUE)

interactions::interact\_plot(logit2, pred = Age, modx = Month10Prop)

interactions::interact\_plot(logit2, pred = Age, modx = GroupProp)

interactions::interact\_plot(logit2, pred =GroupProp, modx = Age)

interactions::interact\_plot(logit2, pred =GroupProp, modx = Month11Prop)

interactions::interact\_plot(logit2, pred =GroupProp, modx = MonthlyAverage, interval = TRUE) #important

#low monthly activity increases the likelihood of churn when group prop goes up, while high activitiy members decrease (mean and over)

interactions::interact\_plot(logit2, pred =GroupProp, modx = Month10Prop)

interactions::interact\_plot(logit2, pred =MonthlyAverage, modx = Month10Prop)

#if the the month12prop is over 1, the member is more active in the last month than over the average for the rest of the year

#the high values of month12prop is estimated as more likely to churn, but

interactions::interact\_plot(logit2, pred = MonthlyAverage, modx = GroupProp)

#the lower the group proportion, the higher the probability of churn

interactions::interact\_plot(logit2, pred = Month10Prop, modx = MonthlyAverage, interval = TRUE)

#the higher the activity in month 12, the higher probability of churn

#when members with low overall average increase their activity in the last month and go beyond their average, their likelihood of churn increases dramatically

#for the values that are either below the member average or about the same, there is little difference in the predicted churn

interactions::interact\_plot(logit2, pred = Month10Prop, modx = MonthlyAverage, interval = TRUE)

#for props placed at the average or below, there is little effect on churn, but when month 10 and 11 experience usage 2 times and above the average, the likelihood of churn skyrockets

#LOWERING THE THRESHOLD VALUE TO CATCH MORE OF THE CHURNING MEMBERS

predict\_prob2<-predict(logit2,newdata = test, type="response")

predicted\_response2<-ifelse(predict\_prob2>0.06,1,0)

predicted\_response2<-as.factor(predicted\_response2)

#CONFUSION MATRIX WITH LOWER THRESHOLD ON THE TEST SET

caret::confusionMatrix(predicted\_response2,test$Churn)

#BINNED RESIDUAL PLOT FOR REDUCED LOGISTIC REGRESSION MODEL

binnedplot(fitted(logit2),

 residuals(logit2, type = "response"),

 nclass = NULL,

 xlab = "Expected Values",

 ylab = "Average Residual",

 main = "Binned Residual Plot: Logit",

 cex.pts = 0.8,

 col.pts = 1,

 col.int = "gray")

#ROC plot for logit model

invisible(plot(roc(train$Churn,

 fitted(logit2)),

 print.auc = T,

 auc.polygon = T,

 print.thres=T,

 col = "blue",

 main = "ROC curve: logit"))

#PROBIT MODEL: STILL BASED ON GLM, BUT WITH CHANGE IN FAMILY TYPE

probit <- glm(Churn ~ . + MonthlyAverage \* GroupProp + MonthlyAverage \* PTProp, family = binomial (link = 'probit'), train)

summary(probit)

tidy(probit)

broom::augment(probit)

confint(probit)

#stepwise removal for a reduced probit model

probit\_2 <- stepAIC(probit) # AIC 1247

probit2 <- glm(Churn ~ MonthlyAverage + Age +

 Private + Month11Prop + Month10Prop

 , family = binomial (link = 'probit'), train)

summary(probit2)

interactions::interact\_plot(probit2, pred = Age, modx = MonthlyAverage)

interactions::interact\_plot(probit2, pred = Age, modx = Month10Prop)

interactions::interact\_plot(probit2, pred = Month10Prop, modx = MonthlyAverage)

interactions::interact\_plot(probit2, pred = GroupProp, modx = Private)

interactions::interact\_plot(probit2, pred = Month10Prop, modx = Age)

interactions::interact\_plot(probit2, pred = MonthlyAverage, modx = Month10Prop)

interactions::interact\_plot(probit2, pred = MonthlyAverage, modx = Age)

#shows coefficient estimates and p-values

tidy(probit2)

#ANALYSIS OF DEVIANCE TABLE: PROBIT

anova(probit2, test = "Chisq")

#when adding variables, the residual deviance drops and the p-values are significant

#which means the variables help explain more of the significant effects on churn

#THE AVERAGE MARGINAL EFFECT OF PROBIT

#dnorm for the normal distribution

ProbitScalar <- mean(dnorm(predict(probit2, type = "link")))

ProbitScalar \* coef(probit2)

#LOWERING THE THRESHOLD VALUE TO CATCH MORE OF THE CHURNING MEMBERS

predict\_prob4<-predict(probit2,newdata = test, type="response")

predicted\_response4<-ifelse(predict\_prob4>0.06,1,0)

predicted\_response4<-as.factor(predicted\_response4)

#CONFUSION MATRIX WITH LOWER THRESHOLD

caret::confusionMatrix(predicted\_response4,test$Churn)

#BINNED RESIDUAL PLOT FOR PROBIT MODEL

binnedplot(fitted(probit2),

 residuals(probit2, type = "response"),

 nclass = NULL,

 xlab = "Expected Values",

 ylab = "Average residual",

 main = "Binned residual plot for Probit Model",

 cex.pts = 0.8,

 col.pts = 1,

 col.int = "gray")

#ROC curve for the probit model

invisible(plot(roc(train$Churn,

 fitted(probit2)),

 print.auc = T,

 auc.polygon = T,

 print.thres=T,

 col = "red",

 main = "ROC curve: probit"))

#install.packages('rattle')

#install.packages('RColorBrewer')

#CLASSIFICATION TREE MODEL

library(rpart)

library(rpart.plot)

library(rattle)

library(RColorBrewer)

#TREE MODEL WITH ALL VARIABLES AVAILABLE

#method = 'class' to initialize binomial classification

#control parameters: minsplit and minbucket forces to create nodes with a minimum size

#trees can't handle interaction terms

tree <- rpart(Churn ~ .,data = train, method = 'class', control = rpart.control(maxdepth = 4 , minsplit = 2, minbucket = 2, cp = -1, prior = c(0.06,0.94)))

#rpart.plot(tree, extra = 106)

# plot mytree

fancyRpartPlot(tree, caption = NULL, palette = "Greys")

#PLOTTING THE TREE MODEL

prp(tree)

rpart.plot(tree)

print(tree)

printcp(tree)

#visualize cross-validation results

plotcp(tree)

summary(tree)

#PRUNE THE TREE

pr.tree <- prune.rpart(tree, cp = -1)

#MAKING PREDICTION ON THE TEST SET

prob\_tree <- predict(tree, newdata = test, type = "prob")[, 2] # Predict class (churn or not)

#CONFUSION MATRIX FOR THE TREE MODEL (ON TEST SET)

predicted\_class1 = factor(ifelse(prob\_tree > 0.06, 1, 0))

caret::confusionMatrix(data = predicted\_class1, reference = test$Churn)

library(ROCR)

library(pROC)

#ROC plot for the tree model

prob\_tree <- ROCR::prediction(predict(tree, type = "prob")[, 2], train$Churn)

plot(performance(prob\_tree, "tpr", "fpr"))

abline(0, 1, lty = 2)