project\_2

Joseph Annand

2024-05-03

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(caret)

## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(writexl)

## Warning: package 'writexl' was built under R version 4.3.3

original\_data <- read.csv("data/Customer\_Dataset\_Data.csv", na.strings = "?", stringsAsFactors = F)  
original\_data <- original\_data %>% column\_to\_rownames(., var = "CustomerID")  
  
data <- original\_data

############################## Data Pre-Processing #############################  
  
# Convert monetary columns from character to numeric  
  
data$VoiceOverTenure <- parse\_number(data$VoiceOverTenure)

## Warning: 3 parsing failures.  
## row col expected actual  
## 842 -- a number #NULL!  
## 2758 -- a number #NULL!  
## 3480 -- a number #NULL!

data$VoiceLastMonth <- parse\_number(data$VoiceLastMonth)  
  
data$EquipmentOverTenure <- parse\_number(data$EquipmentOverTenure)

## Warning: 3296 parsing failures.  
## row col expected actual  
## 3 -- a number $ -  
## 4 -- a number $ -  
## 5 -- a number $ -  
## 7 -- a number $ -  
## 8 -- a number $ -  
## ... ... ........ ......  
## See problems(...) for more details.

data$EquipmentLastMonth <- parse\_number(data$EquipmentLastMonth)

## Warning: 3296 parsing failures.  
## row col expected actual  
## 3 -- a number $ -  
## 4 -- a number $ -  
## 5 -- a number $ -  
## 7 -- a number $ -  
## 8 -- a number $ -  
## ... ... ........ ......  
## See problems(...) for more details.

data$DataOverTenure <- parse\_number(data$DataOverTenure)

## Warning: 3656 parsing failures.  
## row col expected actual  
## 1 -- a number $ -  
## 3 -- a number $ -  
## 4 -- a number $ -  
## 6 -- a number $ -  
## 7 -- a number $ -  
## ... ... ........ ......  
## See problems(...) for more details.

data$DataLastMonth <- parse\_number(data$DataLastMonth)

## Warning: 3656 parsing failures.  
## row col expected actual  
## 1 -- a number $ -  
## 3 -- a number $ -  
## 4 -- a number $ -  
## 6 -- a number $ -  
## 7 -- a number $ -  
## ... ... ........ ......  
## See problems(...) for more details.

data$CardSpendMonth <- parse\_number(data$CardSpendMonth)

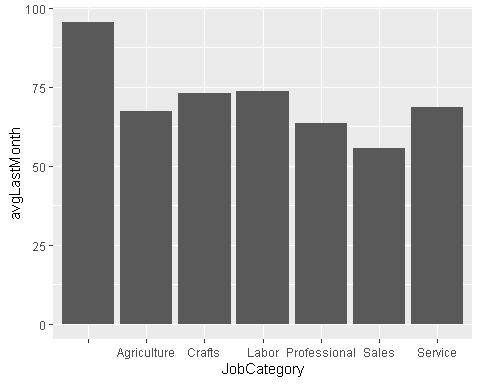
## Warning: 7 parsing failures.  
## row col expected actual  
## 1658 -- a number $ -  
## 1717 -- a number $ -  
## 2800 -- a number $ -  
## 2879 -- a number $ -  
## 4100 -- a number $ -  
## .... ... ........ ......  
## See problems(...) for more details.

data$HHIncome <- parse\_number(data$HHIncome)  
  
# Handle null values in monetary columns  
  
data$VoiceOverTenure <- replace\_na(data$VoiceOverTenure, median(data$VoiceOverTenure))  
data$VoiceLastMonth <- replace\_na(data$VoiceLastMonth, median(data$VoiceLastMonth))  
  
  
data$VoiceOverTenure[is.na(data$VoiceOverTenure)] <- median(data$VoiceOverTenure)  
data$VoiceLastMonth[is.na(data$VoiceLastMonth)] <- median(data$VoiceLastMonth)  
  
handle\_na <- function(data, col1, col2) {  
 med\_val <- median(data[[col1]], na.rm = TRUE) # Calculate median of Column 1  
 for (i in 1:nrow(data)) {  
 if (is.na(data[i, col1])) {  
 if (data[i, col2] == "Yes") {  
 data[i, col1] <- med\_val  
 } else {  
 data[i, col1] <- 0  
 }  
 }  
 }  
 return(data)  
}  
  
data <- handle\_na(data, "EquipmentOverTenure", "EquipmentRental")  
data <- handle\_na(data, "EquipmentLastMonth", "EquipmentRental")  
  
data <- handle\_na(data, "DataOverTenure", "WirelessData")  
data <- handle\_na(data, "DataLastMonth", "WirelessData")  
  
data$CardSpendMonth[is.na(data$CardSpendMonth)] <- median(data$CardSpendMonth, na.rm = T)  
data$TVWatchingHours[is.na(data$TVWatchingHours)] <- mean(data$TVWatchingHours, na.rm = T)  
  
  
# Derived Features  
  
data$TotalOverTenure <- data$VoiceOverTenure + data$EquipmentOverTenure + data$DataOverTenure  
data$TotalByTenure <- data$TotalOverTenure / data$PhoneCoTenure  
data$TotalLastMonth <- data$VoiceLastMonth + data$EquipmentLastMonth + data$DataLastMonth  
  
  
data$TotalServices <- 0  
  
data[2,"EquipmentRental"]

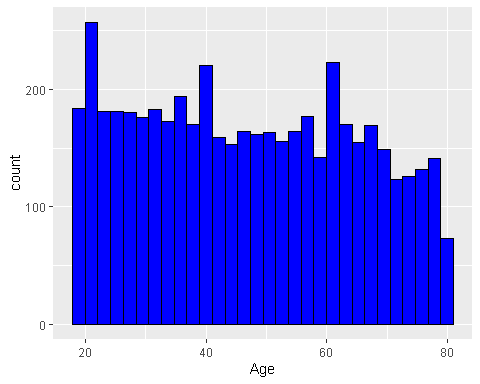
## [1] "Yes"

for (x in (1:nrow(data))) {  
 for (y in c("EquipmentRental","WirelessData","Multiline","VM","Pager","Internet",  
 "CallerID","CallWait","CallForward","ThreeWayCalling")) {  
 if (data[x, y]=="Yes") {  
 data[x, "TotalServices"] <- data[x, "TotalServices"] + 1  
 }  
 }  
}  
  
data$TotalDevices <- 0  
   
for (x in (1:nrow(data))) {  
 for (y in c("Pager","OwnsPC","OwnsMobileDevice","OwnsGameSystem","OwnsFax")) {  
 if (data[x, y]=="Yes") {  
 data[x, "TotalDevices"] <- data[x, "TotalDevices"] + 1  
 }   
 }  
 if (data[x,"TVWatchingHours"] != 0) {  
 data[x, "TotalDevices"] <- data[x, "TotalDevices"] + 1  
 }  
}  
  
# data$LogIncome <- log10(data$HHIncome)  
# data$LogTotalLastMonth <- log10(data$TotalLastMonth)

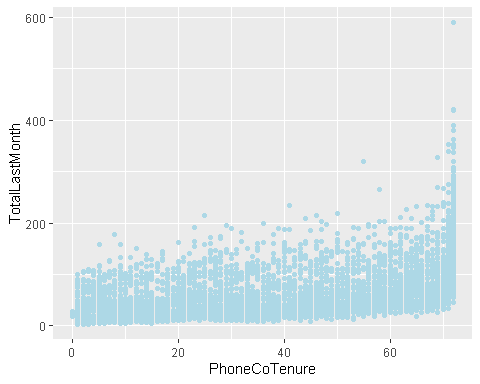
########################### Exploratory Data Analysis ##########################  
  
data\_occupation <- data %>%  
 group\_by(JobCategory) %>%  
 summarise(avgLastMonth = mean(TotalLastMonth))  
  
a <- ggplot(data = data\_occupation) +  
 geom\_col(aes(x=JobCategory, y=avgLastMonth))  
print(a)



b <- ggplot(data = data) +  
 geom\_histogram(aes(x=Age), bins=30, color = "black", fill = "blue")  
print(b)



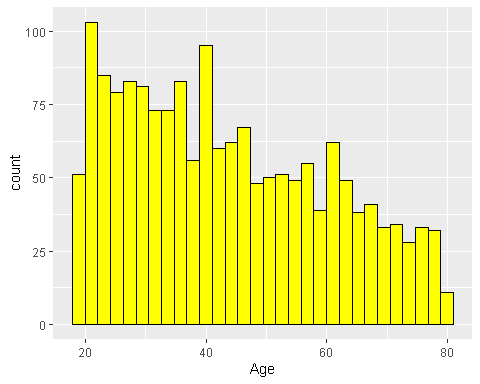
c <- ggplot(data = data) +  
 geom\_point(aes(x=PhoneCoTenure, y=TotalLastMonth), color = "lightblue")  
print(c)



data\_equip <- data %>%  
 add\_count(EquipmentRental) %>%  
 group\_by(EquipmentRental,n) %>%  
 summarise(avgLastMonth = mean(TotalLastMonth))

## `summarise()` has grouped output by 'EquipmentRental'. You can override using  
## the `.groups` argument.

d <- ggplot(data = data %>% filter(EquipmentRental == "Yes")) +  
 geom\_histogram(aes(x=Age), color="black", fill="yellow", bins=30)  
print(d)

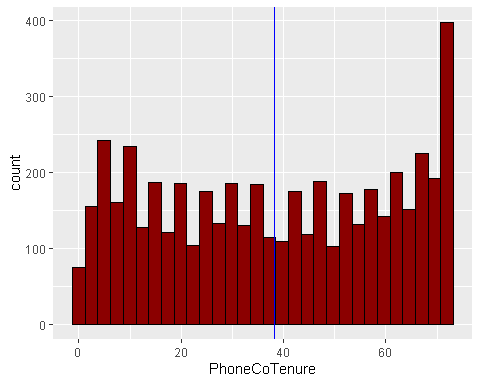


data\_wireless <- data %>%  
 add\_count(WirelessData) %>%  
 group\_by(WirelessData,n) %>%  
 summarise(avgLastMonth = mean(TotalLastMonth))

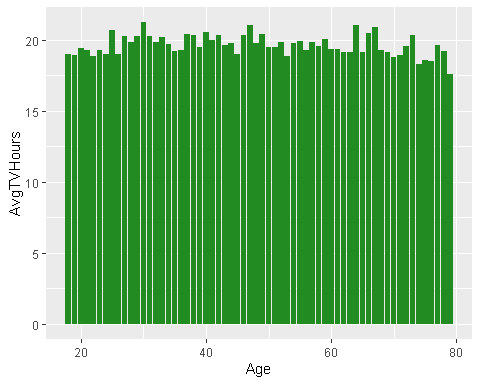
## `summarise()` has grouped output by 'WirelessData'. You can override using the  
## `.groups` argument.

# data <- data %>% mutate(segment = case\_when(  
# EquipmentRental == "Yes" & WirelessData == "Yes" ~ 1,  
# EquipmentRental == "Yes" & WirelessData == "No" ~ 2,  
# EquipmentRental == "No" & WirelessData == "Yes" ~ 3,  
# EquipmentRental == "No" & WirelessData == "No" ~ 4,  
# ))  
#   
# data\_segmented <- data %>% add\_count(segment) %>%  
# group\_by(segment,n) %>%  
# select\_if(is.numeric) %>%  
# summarise\_all("median")  
  
e <- ggplot(data = data) +  
 geom\_histogram(aes(x=PhoneCoTenure), color="black", fill="darkred") +  
 geom\_vline(xintercept = mean(data$PhoneCoTenure), color="blue")  
print(e)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# f <- ggplot(data = data) +  
# geom\_point(aes(x=LogIncome, y=LogTotalLastMonth), color="blue")  
# print(f)  
  
# cor(data$LogIncome, data$LogTotalLastMonth)  
  
g <- ggplot(data = data %>% select(Age, TVWatchingHours) %>%  
 group\_by(Age) %>%  
 summarise(AvgTVHours = mean(TVWatchingHours))) +  
 geom\_col(aes(x=Age, y=AvgTVHours), fill="forestgreen")  
print(g)

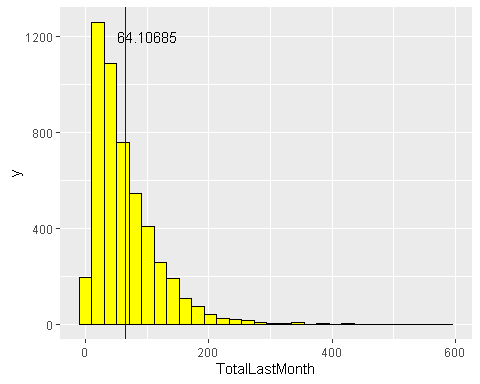


# missing\_cols <- is.na(data$PhoneCoTenure) | is.na(data$CardSpendMonth) | is.na(data$TotalLastMonth)  
#   
# data <- na.omit(data, subset = !missing\_cols)  
  
  
summary(data %>% select(PhoneCoTenure, TotalLastMonth, CardSpendMonth))

## PhoneCoTenure TotalLastMonth CardSpendMonth   
## Min. : 0.0 Min. : 2.85 Min. : 69.7   
## 1st Qu.:18.0 1st Qu.: 26.70 1st Qu.: 1839.8   
## Median :38.0 Median : 49.73 Median : 2766.9   
## Mean :38.2 Mean : 64.11 Mean : 3375.9   
## 3rd Qu.:59.0 3rd Qu.: 87.30 3rd Qu.: 4185.4   
## Max. :72.0 Max. :590.40 Max. :39264.1

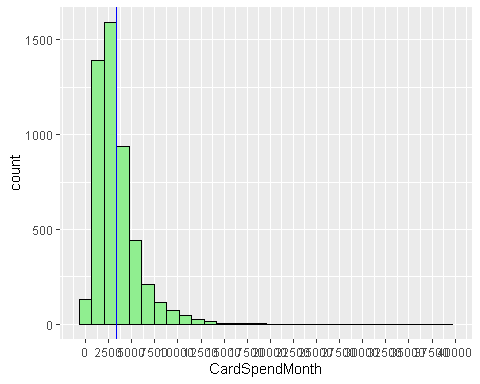
###################### Rule Based Segmentation #################################  
  
h <- ggplot(data = data) +  
 geom\_histogram(aes(x=TotalLastMonth), color="black", fill="yellow") +  
 geom\_vline(xintercept=mean(data$TotalLastMonth), color="blue") +  
 annotate(geom="text", x=100, y=1200, label=as.character(mean(data$TotalLastMonth)))  
print(h)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



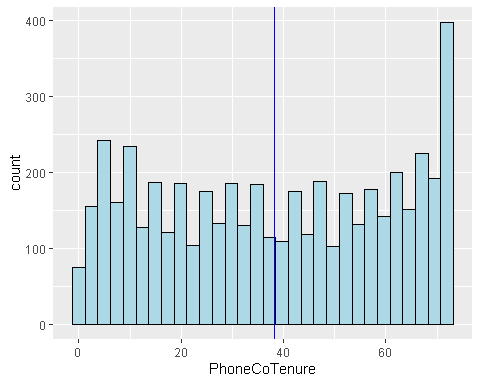
data$LastMonthGroup <- cut(data$TotalLastMonth, breaks=c(-1,70,600), labels=c("Low", "High"))  
  
i <- ggplot(data = data) +  
 geom\_histogram(aes(x=CardSpendMonth), color="black", fill="lightgreen") +  
 geom\_vline(xintercept=mean(data$CardSpendMonth, na.rm = T), color="blue") +  
 scale\_x\_continuous(breaks = seq(0, 40000, by=2500))  
print(i)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



data$CardSpendGroup <- cut(data$CardSpendMonth, breaks=c(-1,5000,40000), labels=c("Low", "High"))  
  
j <- ggplot(data = data) +  
 geom\_histogram(aes(x=PhoneCoTenure), color="black", fill="lightblue") +  
 geom\_vline(xintercept=mean(data$PhoneCoTenure), na.rm = T, color="blue")  
print(j)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



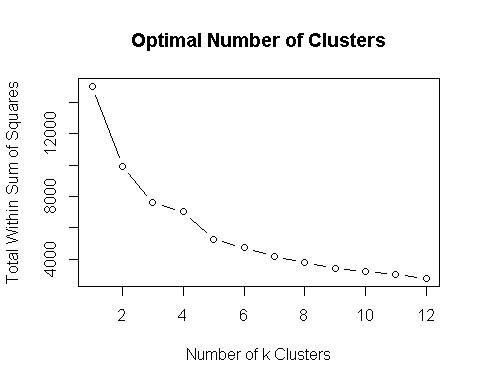
data$TenureGroup <- cut(data$PhoneCoTenure, breaks=c(-1,40,100), labels=c("Short", "Long"))  
  
  
data <- data %>% mutate(segment = case\_when(  
 LastMonthGroup == "Low" & CardSpendGroup == "Low" & TenureGroup == "Short" ~ 1,  
 LastMonthGroup == "Low" & CardSpendGroup == "Low" & TenureGroup == "Long" ~ 2,  
 LastMonthGroup == "Low" & CardSpendGroup == "High" & TenureGroup == "Short" ~ 3,  
 LastMonthGroup == "Low" & CardSpendGroup == "High" & TenureGroup == "Long" ~ 4,  
 LastMonthGroup == "High" & CardSpendGroup == "Low" & TenureGroup == "Short" ~ 5,  
 LastMonthGroup == "High" & CardSpendGroup == "Low" & TenureGroup == "Long" ~ 6,  
 LastMonthGroup == "High" & CardSpendGroup == "High" & TenureGroup == "Short" ~ 7,  
 LastMonthGroup == "High" & CardSpendGroup == "High" & TenureGroup == "Long" ~ 8  
))  
  
data$CustomerValue <- ifelse(data$segment %in% c(4,6,7,8),"High","Low")  
data$CustomerValue <- as.factor(data$CustomerValue)

###################### Supervised Learning Segmentation ########################  
  
  
log\_reg <- glm(CustomerValue ~ TotalLastMonth + CardSpendMonth + PhoneCoTenure, data=data,  
 family = "binomial")

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

data$probValue <- predict(log\_reg, newdata = data, type="response")  
data$ValueGroup <- cut(data$probValue, breaks=c(0.0,0.50,1.0),  
 labels=c("Low","High"))

###################### Unsupervised Learning Segmentation ######################  
  
# Standardize numeric variables  
k\_points <- data %>% select(c("PhoneCoTenure", "TotalLastMonth", "CardSpendMonth"))  
k\_points <- scale(k\_points)  
  
  
  
# Select best number of k-values  
ks <- 1:12  
tot\_within\_ss <- sapply(ks, function(k) {  
 set.seed(1223)  
 cl <- kmeans(k\_points, k)  
 cl$tot.withinss  
})  
plot(ks, tot\_within\_ss, type = "b", ylab = "Total Within Sum of Squares",  
 xlab = "Number of k Clusters", main = "Optimal Number of Clusters")



set.seed(1223)  
NUM\_CLUSTERS <- 3  
kclust <- kmeans(k\_points, centers = NUM\_CLUSTERS, nstart=10)  
  
#add segments to original dataset  
data$kmeans\_segment <- as.factor(kclust$cluster)