HW2

Mikayla Norton

2023-01-21

Problem 1

The purpose of this problem is to create a modeling dataset for customer churn in the next year, based on the customer history dataset. In ICA 3, you performed the row augmentation to get missing rows added to the data. To get to the modeling dataset:

Part A

Create the target field based on the strategy given in the slides.

```
customer_history <- read.csv("orders/Customer_History.csv")
years <- data.frame(year = seq(2005,2022))

customer_history2 <- sqldf("SELECT *
               FROM customer_history
               INNER JOIN years
               WHERE year <= Last_Year
               AND year >= First_Year")
head(customer_history2)
```

```
##   Customer First_Year Last_Year Year_Born Male_Female year
## 1        1       2007      2017      1961           F 2007
## 2        1       2007      2017      1961           F 2008
## 3        1       2007      2017      1961           F 2009
## 4        1       2007      2017      1961           F 2010
## 5        1       2007      2017      1961           F 2011
## 6        1       2007      2017      1961           F 2012
```

Part B

Create the following additional input variables that would be used to make predictions: i. Customer age

```
customer_history2 <- sqldf("SELECT *, year-Year_born as Age
               FROM customer_history2")

head(customer_history2)
```

```
##   Customer First_Year Last_Year Year_Born Male_Female year Age
## 1        1       2007      2017      1961           F 2007  46
## 2        1       2007      2017      1961           F 2008  47
## 3        1       2007      2017      1961           F 2009  48
## 4        1       2007      2017      1961           F 2010  49
## 5        1       2007      2017      1961           F 2011  50
## 6        1       2007      2017      1961           F 2012  51
```

ii. Number of years as a customer

```
customer_history2 <- sqldf("SELECT *, Last_Year-First_Year as Years_As_Cust
               FROM customer_history2")
head(customer_history2)
```

```
##   Customer First_Year Last_Year Year_Born Male_Female year Age Years_As_Cust
## 1        1       2007      2017      1961           F 2007  46            10
## 2        1       2007      2017      1961           F 2008  47            10
## 3        1       2007      2017      1961           F 2009  48            10
## 4        1       2007      2017      1961           F 2010  49            10
## 5        1       2007      2017      1961           F 2011  50            10
## 6        1       2007      2017      1961           F 2012  51            10
```

iii. Using the complaints dataset, create a count of complaints made each year and join it to the dataset (be careful with what year you join!)

```
complaints <- read.csv("orders/Complaint_History.csv")

complaintsSummary <- sqldf("SELECT CustomerID, ComplaintYear,COUNT(*) as ComplaintCount
                           FROM complaints
                           GROUP BY ComplaintYear, CustomerID
                           ORDER BY ComplaintYear")

Cust_Complaints <- sqldf("SELECT customer_history2.*,ComplaintCount
                         FROM customer_history2
                         LEFT JOIN complaintsSummary
                         ON customer_history2.year = complaintsSummary.ComplaintYear
                         AND customer_history2.Customer=complaintsSummary.CustomerID")
Cust_Complaints["ComplaintCount"][is.na(Cust_Complaints["ComplaintCount"])] <- 0

head(Cust_Complaints)
```

```
##   Customer First_Year Last_Year Year_Born Male_Female year Age Years_As_Cust
## 1        1       2007      2017      1961           F 2007  46            10
## 2        1       2007      2017      1961           F 2008  47            10
## 3        1       2007      2017      1961           F 2009  48            10
## 4        1       2007      2017      1961           F 2010  49            10
## 5        1       2007      2017      1961           F 2011  50            10
## 6        1       2007      2017      1961           F 2012  51            10
##   ComplaintCount
## 1              1
## 2              0
## 3              0
## 4              0
## 5              1
## 6              0
```

Problem 2

The purpose of this problem is to create a modeling dataset for predicting which orders will order which products, based on the order history dataset. In ICA 3, you performed the row augmentation. Now:

Part A

Create the target field based on the strategy given in the slides.

```
order_history <- read.csv("orders/Order_History.csv")
order_history2 <- sqldf("SELECT CustomerID, Product_ID, Year, Month, sum(Quantity) as Quantity
                        FROM order_history
                        GROUP BY customerID, Product_ID, Year, Month
                        ORDER BY CustomerID, Product_ID, Year, Month")

order_wide <- pivot_wider(order_history2, names_from = Year, values_from = Quantity)
order_long <- pivot_longer(order_wide, cols = starts_with('20'), names_to = 'Year', values_to = 'Quanti
order_long <- sqldf("SELECT *
                    FROM order_long
                    ORDER BY  CustomerID, Product_ID, Year, Month")

order_long["Quantity"][is.na(order_long["Quantity"])] <- 0
order_long["Quantity"][order_long["Quantity"] != 0] <- 1

order_long <- order_long %>%
       rename("OrderBinary" = "Quantity")

head(order_long)
```

```
##   CustomerID Product_ID Month Year OrderBinary
## 1          1          1     1 2008           0
## 2          1          1     2 2008           1
## 3          1          1     3 2008           0
## 4          1          1     4 2008           0
## 5          1          1     6 2008           0
## 6          1          1     7 2008           0
```

```
year <- data.frame("year" = min(order_long$Year):max(order_long$Year))
month <- data.frame("month" = min(order_long$Month):max(order_long$Month))
cust <- data.frame("customer" = min(order_long$CustomerID):max(order_long$CustomerID))
products <- data.frame("product" = min(order_long$Product_ID):max(order_long$Product_ID))

cartesian_orders <- sqldf("SELECT *
                    FROM year
                    INNER JOIN month
                    INNER JOIN cust
                    INNER JOIN products")

# order_summary <- sqldf("SELECT *,
#                        CASE WHEN Month = 1 THEN 12 ELSE Month-1 END AS PastMonth
#                        FROM order_long
#                        GROUP BY CustomerID, Year, Month")
#
# order_summary <- sqldf("SELECT *,
#                        CASE WHEN Month = 1 THEN Year-1 ELSE Year END AS PastYear
#                        FROM order_summary
#                        GROUP BY CustomerID, Year, Month")
# head(order_summary)
```

Part B

Create the following additional input variables that would be used to make predictions. All of these x variables should be from the standpoint of the prior month.

i. Number of months since the last order

```
Order_summaries <- sqldf("SELECT customer, product, t1.year, t1.month, SUM(OrderBinary) as TotalOrders
                         FROM cartesian_orders t1
                         LEFT JOIN order_long t2
                         ON t1.customer=t2.CustomerID
                         AND t1.product=t2.Product_ID
                         AND t1.year=t2.Year
                         AND t1.month=t2.Month
                         GROUP BY customer, product, t1.year, t1.month")

Order_summaries["TotalOrders"][is.na(Order_summaries["TotalOrders"])] <- 0

#####


Months_Since <- sqldf("SELECT t1.month, t1.year, t1.customer,t1.product, t1.TotalOrders, (t1.year*12+t1
                         FROM Order_summaries t1
                         INNER JOIN Order_summaries t2
                         ON t1.customer=t2.customer
                         AND t1.product=t2.product
                         WHERE t1.TotalOrders>0 AND t2.TotalOrders>0 AND MonthsSinceLast > 0 AND Months

Months_Since_Last <- sqldf("SELECT customer, product, month, year, min(MonthsSinceLast) as MonthsSinceLa
                         FROM Months_Since
                         GROUP BY customer, product, year, month
                         ORDER BY customer, product, year, month")
head(Months_Since_Last)
```

```
##   customer product month year MonthsSinceLast
## 1        1       1     9 2008               7
## 2        1       1    10 2008               1
## 3        1       1     7 2009               9
## 4        1       1     8 2009               1
## 5        1       1     9 2009               1
## 6        1       1     2 2010               5
```

ii. Number of months out of the previous 12 months that an order was placed

```
Previous12 <- sqldf("SELECT customer, product, month, year, COUNT(MonthsSinceLast) as OrdersOverLastYea
                    FROM Months_Since
                    GROUP BY customer, product, year, month
                    ORDER BY customer, product, year, month")
head(Previous12)
```

```
##   customer product month year OrdersOverLastYear
## 1        1       1     9 2008                  1
## 2        1       1    10 2008                  2
## 3        1       1     7 2009                  2
## 4        1       1     8 2009                  3
## 5        1       1     9 2009                  4
## 6        1       1     2 2010                  3
```

iii. Average quantity ordered per order (over last 12 years)

```
Avg_Quantity <- sqldf("SELECT t1.Month, t1.Year, t1.CustomerID, t1.Product_ID, avg(t2.Quantity) as AvgQ
                         FROM order_history2 t1
```

```
                        INNER JOIN order_history2 t2
                        ON t1.CustomerID=t2.CustomerID
                        AND t1.Product_ID=t2.Product_ID
                        WHERE t1.Quantity>0 AND t2.Quantity>0 AND MonthsSinceLast >= 0 AND MonthsSincel
                        GROUP BY t1.CustomerID, t1.Product_ID, t1.Year, t1.Month
                        ORDER BY t1.CustomerID, t1.Product_ID, t1.Year, t1.Month")

Avg_Quantity <- sqldf("SELECT CustomerID, Product_ID, Month, Year, AvgQuantity
                      FROM Avg_Quantity
                      ORDER BY CustomerID, Product_ID, Year, Month")
head(Avg_Quantity)
```

```
##   CustomerID Product_ID Month Year AvgQuantity
## 1          1          1     2 2008    69.00000
## 2          1          1     9 2008   109.00000
## 3          1          1    10 2008    95.00000
## 4          1          1     7 2009    79.66667
## 5          1          1     8 2009    80.25000
## 6          1          1     9 2009    66.40000
```

Problem 3

With the OJ dataset, calculate some additional X's and join them back to the original dataset; specifically:

```
oj <- ISLR2::OJ
head(oj)
```

```
##   Purchase WeekofPurchase StoreID PriceCH PriceMM DiscCH DiscMM SpecialCH
## 1       CH            237       1    1.75    1.99   0.00    0.0         0
## 2       CH            239       1    1.75    1.99   0.00    0.3         0
## 3       CH            245       1    1.86    2.09   0.17    0.0         0
## 4       MM            227       1    1.69    1.69   0.00    0.0         0
## 5       CH            228       7    1.69    1.69   0.00    0.0         0
## 6       CH            230       7    1.69    1.99   0.00    0.0         0
##   SpecialMM  LoalCH SalePriceMM SalePriceCH PriceDiff Store7 PctDiscMM
## 1         0 0.500000        1.99        1.75      0.24     No  0.000000
## 2         1 0.600000        1.69        1.75     -0.06     No  0.150754
## 3         0 0.680000        2.09        1.69      0.40     No  0.000000
## 4         0 0.400000        1.69        1.69      0.00     No  0.000000
## 5         0 0.956535        1.69        1.69      0.00    Yes  0.000000
## 6         1 0.965228        1.99        1.69      0.30    Yes  0.000000
##   PctDiscCH ListPriceDiff STORE
## 1  0.000000          0.24     1
## 2  0.000000          0.24     1
## 3  0.091398          0.23     1
## 4  0.000000          0.00     1
## 5  0.000000          0.00     0
## 6  0.000000          0.30     0
```

Part A

The average price for each brand by store

```
ojAvgPrice <- sqldf("SELECT *, avg(PriceCH) as AvgPriceCH, avg(PriceMM) as AvgPriceMM
             FROM oj
```

```
            GROUP BY StoreID")
head(ojAvgPrice)
```

```
##   Purchase WeekofPurchase StoreID PriceCH PriceMM DiscCH DiscMM SpecialCH
## 1       CH            237       1    1.75    1.99    0.0    0.0         0
## 2       MM            268       2    1.86    2.18    0.0    0.0         0
## 3       CH            251       3    1.99    2.23    0.0    0.0         0
## 4       CH            271       4    1.99    2.09    0.1    0.4         1
## 5       CH            228       7    1.69    1.69    0.0    0.0         0
##   SpecialMM  LoyalCH SalePriceMM SalePriceCH PriceDiff Store7 PctDiscMM
## 1         0 0.500000        1.99        1.75      0.24     No  0.000000
## 2         1 0.400000        2.18        1.86      0.32     No  0.000000
## 3         0 0.544000        2.23        1.99      0.24     No  0.000000
## 4         0 0.400000        1.69        1.89     -0.20     No  0.191388
## 5         0 0.956535        1.69        1.69      0.00    Yes  0.000000
##   PctDiscCH ListPriceDiff STORE AvgPriceCH AvgPriceMM
## 1  0.000000          0.24     1   1.803758   2.022102
## 2  0.000000          0.32     2   1.841216   2.073198
## 3  0.000000          0.24     3   1.928265   2.127041
## 4  0.050251          0.10     4   1.954029   2.108417
## 5  0.000000          0.00     0   1.844522   2.089045
```

Part B

The previous week's fraction of customers who bought Minute Maid, by store (note: you will lose the first week when you use this as an X)

```
ojCustCounts <- sqldf("SELECT *, COUNT(*) as TotalPurchases,
                      COUNT(CASE WHEN Purchase = 'MM' THEN 1 ELSE NULL END) AS CountMM
                      FROM oj
                      GROUP BY WeekofPurchase")

ojCustCounts$TotalPurchases <- as.double(ojCustCounts$TotalPurchases)
ojCustCounts$CountMM <- as.double(ojCustCounts$CountMM)

ojFrac <- sqldf("SELECT t1.*, (t2.CountMM)/(t2.TotalPurchases) as CustFrac
                FROM ojCustCounts t1
                INNER JOIN ojCustCounts t2
                WHERE t1.WeekofPurchase = t2.WeekofPurchase+1")




head(ojFrac)
```

```
##   Purchase WeekofPurchase StoreID PriceCH PriceMM DiscCH DiscMM SpecialCH
## 1       CH            228       7    1.69    1.69      0    0.0         0
## 2       CH            229       4    1.79    1.79      0    0.0         0
## 3       CH            230       7    1.69    1.99      0    0.0         0
## 4       CH            231       3    1.79    1.79      0    0.0         0
## 5       CH            232       7    1.69    1.99      0    0.4         1
## 6       CH            233       4    1.79    2.09      0    0.0         0
##   SpecialMM  LoyalCH SalePriceMM SalePriceCH PriceDiff Store7 PctDiscMM
## 1         0 0.956535        1.69        1.69       0.0    Yes  0.000000
## 2         0 0.936769        1.79        1.79       0.0     No  0.000000
```

```
## 3         1 0.965228        1.99        1.69       0.3    Yes  0.000000
## 4         0 0.895142        1.79        1.79       0.0     No  0.000000
## 5         1 0.972182        1.59        1.69      -0.1    Yes  0.201005
## 6         0 0.959532        2.09        1.79       0.3     No  0.000000
##   PctDiscCH ListPriceDiff STORE TotalPurchases CountMM  CustFrac
## 1         0           0.0     0             22      13 0.6000000
## 2         0           0.0     4             23      14 0.5909091
## 3         0           0.3     0             19       8 0.6086957
## 4         0           0.0     3             20      11 0.4210526
## 5         0           0.3     0             19       7 0.5500000
## 6         0           0.3     4             28      10 0.3684211
```