Alexandra DeGrandchamp
November 19, 2022

Homework 5

Data Gathering and Integration

I chose to work with the Customer Personality Analysis found on Kaggle (https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis?resource=download). This data set stems from a customer survey of "ideal" customers and features many quantitative and qualitative variables on 2,240 customer records. I chose to use this data set because I thought it would lend itself well to both clustering (finding clusters or segments of customers is a useful tool for companies, and a common analyst ask) as well as classification. Exactly WHAT to classify wasn't immediately obvious from the data set, so before starting, I did a little data manipulation to find distributions of customer characteristics to explore interesting classification possibilities.

```
campaignFeatures <- campaign %>%
                        mutate(boughtWine = if_else(MntWines > 0, 1, 0),
                               boughtFruit = if_else(MntFruits > 0, 1, 0),
                               boughtMeat = if_else(MntMeatProducts > 0, 1, 0),
                               boughtFish = if_else(MntFishProducts > 0, 1, 0),
                               boughtSweet = if_else(MntSweetProducts > 0, 1, 0
),
                               boughtGold = if_else(MntGoldProds > 0, 1, 0),
                               webPurchaser = if_else(NumWebPurchases > 0, 1, 0
),
                               catPurchaser = if_else(NumCatalogPurchases > 0,
1, 0),
                               storePurchaser = if_else(NumStorePurchases > 0,
1, 0)) %>%
                        select(boughtWine, boughtFruit, boughtMeat, boughtFish,
boughtSweet,
                               boughtGold, webPurchaser, catPurchaser, storePur
chaser) %>%
                        pivot_longer(everything(),names_to = "characteristic",
                                     values_to = "sum") %>%
                        group_by(characteristic) %>%
                        summarize(sum = sum(sum),
                                  pct = (sum/2240)*100)
campaignFeatures
```

```
## # A tibble: 9 × 3
##   characteristic   sum   pct
##   <chr>          <dbl> <dbl>
## 1 boughtFish      1856  82.9
## 2 boughtFruit     1840  82.1
## 3 boughtGold      2179  97.3
## 4 boughtMeat      2239 100.
## 5 boughtSweet     1821  81.3
## 6 boughtWine      2227  99.4
## 7 catPurchaser    1654  73.8
```

```
## 8 storePurchaser  2225  99.3
## 9 webPurchaser    2191  97.8
```

I wanted to choose a classification variable that was binary and not too heavily weighted toward one of the two categories. For this exercise, I'll be classifying if the customer is someone who has made a Catalog Purchase (in the table above, catPurchaser represents those who have purchased from a catalog; 73.84% of customers surveyed in this data set had). Understanding how customers shop is a key insight for retailers, and catalogs can be an expensive and wasteful marketing tool if customers won't buy from a catalog. This classifier could be used to best target new customers or prospective customers with the right marketing technique: sending catalogs to those who appreciate them, skipping those who are unlikely to purchase. This classification method could later be expanded to store and web purchasing with a few tweaks, since those are more heavily weighted in favor of "yes" in this data set.

Once I selected the tasks I would perform, it was time to explore the data.

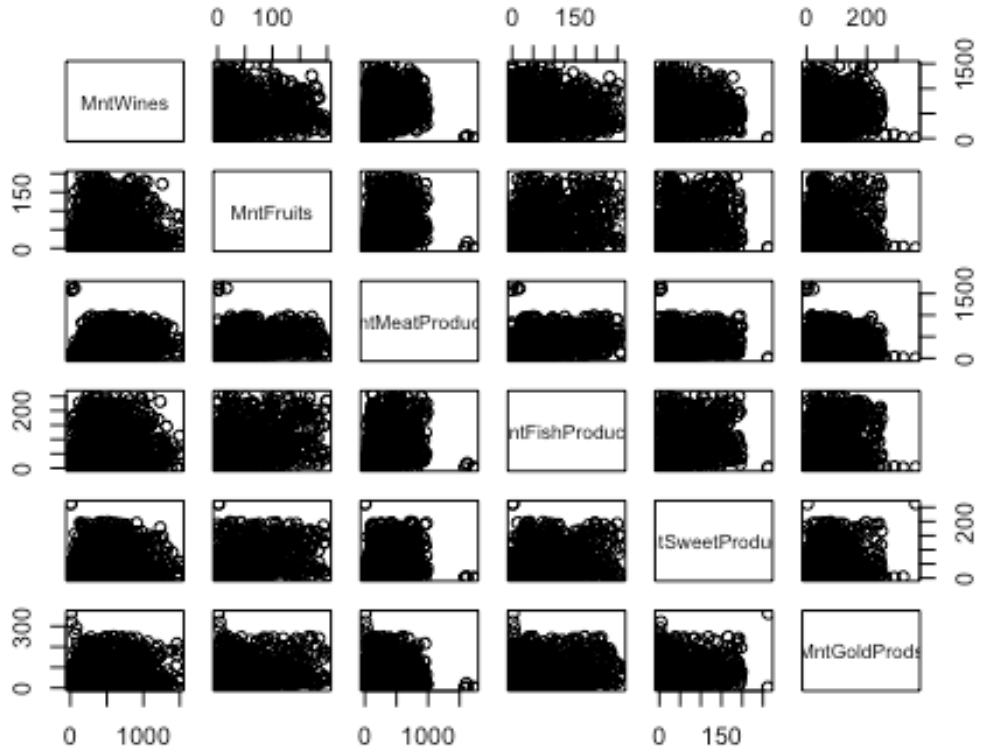
Data Exploration

To start exploring the data, I observed the summary of the data set:

```
##        ID          Year_Birth        Education     Marital_Status
##  Min.   :    0   Min.   :1893   2n Cycle  : 203   Married :864
##  1st Qu.: 2828   1st Qu.:1959   Basic     :  54   Together:580
##  Median : 5458   Median :1970   Graduation:1127   Single  :480
##  Mean   : 5592   Mean   :1969   Master    : 370   Divorced:232
##  3rd Qu.: 8428   3rd Qu.:1977   PhD       : 486   Widow   : 77
##  Max.   :11191   Max.   :1996                     Alone   :  3
##                                                   (Other) :  4
##      Income          Kidhome          Teenhome         Dt_Customer
##  Min.   :  1730   Min.   :0.0000   Min.   :0.0000   31-08-2012:  12
##  1st Qu.: 35303   1st Qu.:0.0000   1st Qu.:0.0000   12-05-2014:  11
##  Median : 51382   Median :0.0000   Median :0.0000   12-09-2012:  11
##  Mean   : 52247   Mean   :0.4442   Mean   :0.5062   14-02-2013:  11
##  3rd Qu.: 68522   3rd Qu.:1.0000   3rd Qu.:1.0000   20-08-2013:  10
##  Max.   :666666   Max.   :2.0000   Max.   :2.0000   22-05-2014:  10
##  NA's   :24                                         (Other)   :2175
##     Recency         MntWines         MntFruits     MntMeatProducts
##  Min.   : 0.00   Min.   :   0.00   Min.   :  0.0   Min.   :   0
##  1st Qu.:24.00   1st Qu.:  23.75   1st Qu.:  1.0   1st Qu.:  16
##  Median :49.00   Median : 173.50   Median :  8.0   Median :  67
##  Mean   :49.11   Mean   : 303.94   Mean   : 26.3   Mean   : 167
##  3rd Qu.:74.00   3rd Qu.: 504.25   3rd Qu.: 33.0   3rd Qu.: 232
##  Max.   :99.00   Max.   :1493.00   Max.   :199.0   Max.   :1725
##
##  MntFishProducts  MntSweetProducts  MntGoldProds    NumDealsPurchases
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  3.00   1st Qu.:  1.00   1st Qu.:  9.00   1st Qu.: 1.000
##  Median : 12.00   Median :  8.00   Median : 24.00   Median : 2.000
##  Mean   : 37.53   Mean   : 27.06   Mean   : 44.02   Mean   : 2.325
##  3rd Qu.: 50.00   3rd Qu.: 33.00   3rd Qu.: 56.00   3rd Qu.: 3.000
##  Max.   :259.00   Max.   :263.00   Max.   :362.00   Max.   :15.000
##
```

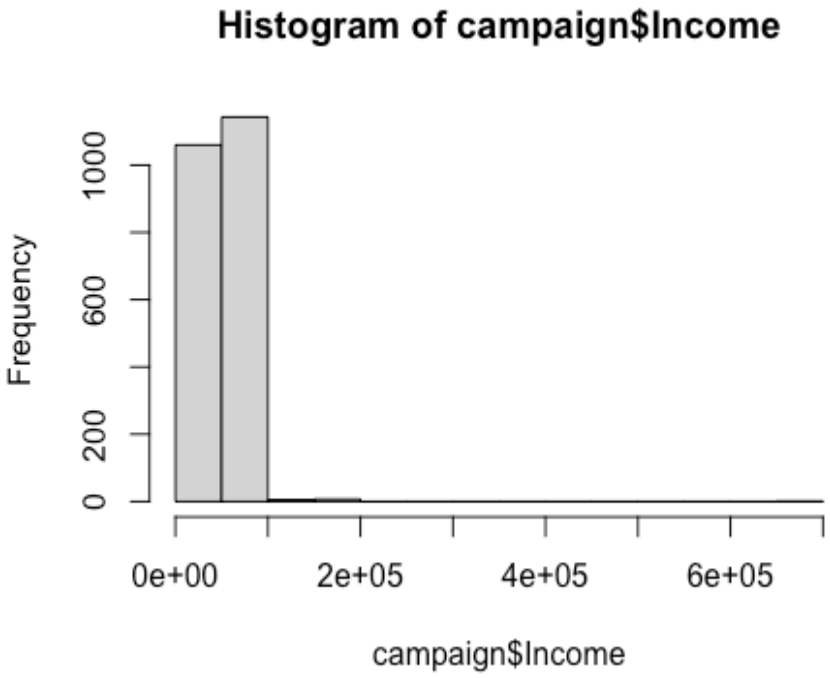
```
##  NumWebPurchases  NumCatalogPurchases NumStorePurchases NumWebVisitsMonth
##  Min.   : 0.000   Min.   : 0.000      Min.   : 0.00     Min.   : 0.000
##  1st Qu.: 2.000   1st Qu.: 0.000      1st Qu.: 3.00     1st Qu.: 3.000
##  Median : 4.000   Median : 2.000      Median : 5.00     Median : 6.000
##  Mean   : 4.085   Mean   : 2.662      Mean   : 5.79     Mean   : 5.317
##  3rd Qu.: 6.000   3rd Qu.: 4.000      3rd Qu.: 8.00     3rd Qu.: 7.000
##  Max.   :27.000   Max.   :28.000      Max.   :13.00     Max.   :20.000
##
##   AcceptedCmp3      AcceptedCmp4      AcceptedCmp5      AcceptedCmp1
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.07277   Mean   :0.07455   Mean   :0.07277   Mean   :0.06429
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##   AcceptedCmp2        Complain        Z_CostContact   Z_Revenue
##  Min.   :0.00000   Min.   :0.000000   Min.   :3     Min.   :11
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:3     1st Qu.:11
##  Median :0.00000   Median :0.000000   Median :3     Median :11
##  Mean   :0.01339   Mean   :0.009375   Mean   :3     Mean   :11
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:3     3rd Qu.:11
##  Max.   :1.00000   Max.   :1.000000   Max.   :3     Max.   :11
##
##     Response
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1491
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

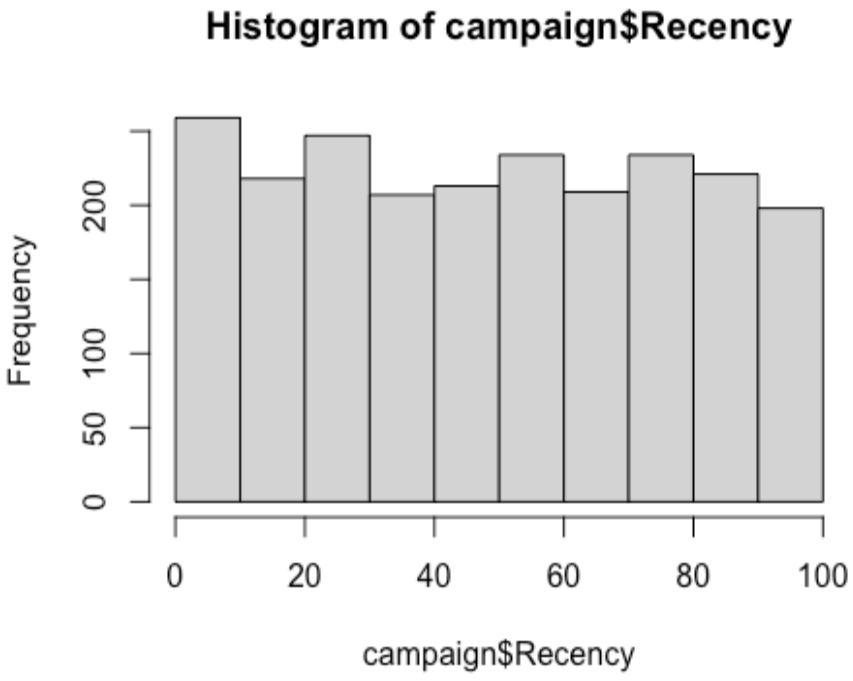
There are a few early outliers in Year_Birth, Marital_Status, and Income, as well as some N/A data in Income Z_CostContact and Z_Revenue won't be useful due to having identical values for each observation. Since there's a large number of numeric values, it's useful to explore correlations. I observed correlations in dollar value of purchases per category.



This correlation seems to suggest that buying meat products correlates most strongly to purchasing other types of products, showing a positive, often fanned relationship. This could suggest that meat buyers may be more amenable to add-on purchases or that meat is an enticing add-on suggestion.

Categorical variables were explored with histograms or bar graphs of counts. Most interesting were the distribution of income (one, an outlier with income level listed as $666,666, was removed) and recency. For income, a relatively bell-shaped distribution exists, but there is a long tail as incomes trend higher. For recency, the data shows an even distribution.





The data exploration provided some early insights into the data set, as well as some stark cleaning needs.

Data Cleaning

I performed a variety of data cleaning tasks, such as:

- removing rows that had NULL values (there were only 24 in a data set of 2200+)
- removing row with outlier/incorrect income value
- removing rows with outlier/incorrect Year_Birth values (3 total); I also calculated an age variable based on surmised data set date of 2014
- re-factoring and cleaning nonsense answers on Marital_Status; transformed to relationshipStatus and removed redundant Marital_Status
- removing unneeded variables ID, Z_CostContact, Z_Revenue
- re-factoring and cleaning nonsense answers on Education; transformed to educationLevel and removed redundant Education
- creating binary variable hasChildren
- creating numeric variable numPurchases (sum of all purchases, last 2 years)
- creating numeric variable amtSpent (sum of all purchase totals, last 2 years)
- transforming date into usable date object as customerFromDate, extracting yearCust as a factor variable, creating custLength as measure of months (a rough estimation; assumed 4-week months) spent as customer
- confirming that the AcceptedCmpX variables (5 total) were true binary variables (in other words, confirmed that AcceptedCmp1 did not exclude a customer from AcceptedCmp2)

The clean data set and its summary statistics are below:

```
summary(campaignClean)

##    Year_Birth       Income          Kidhome          Teenhome
##  1976   :  89   Min.   :  1730   Min.   :0.0000   Min.   :0.0000
##  1971   :  86   1st Qu.: 35246   1st Qu.:0.0000   1st Qu.:0.0000
##  1975   :  83   Median : 51373   Median :0.0000   Median :0.0000
##  1972   :  78   Mean   : 52237   Mean   :0.4419   Mean   :0.5056
##  1978   :  76   3rd Qu.: 68487   3rd Qu.:1.0000   3rd Qu.:1.0000
##  1970   :  75   Max.   :666666   Max.   :2.0000   Max.   :2.0000
##  (Other):1726
##     Recency         MntWines        MntFruits      MntMeatProducts
##  Min.   : 0.00   Min.   :   0.0   Min.   :  0.00   Min.   :   0
##  1st Qu.:24.00   1st Qu.:  24.0   1st Qu.:  2.00   1st Qu.:  16
##  Median :49.00   Median : 175.0   Median :  8.00   Median :  68
##  Mean   :49.01   Mean   : 305.2   Mean   : 26.32   Mean   : 167
##  3rd Qu.:74.00   3rd Qu.: 505.0   3rd Qu.: 33.00   3rd Qu.: 232
##  Max.   :99.00   Max.   :1493.0   Max.   :199.00   Max.   :1725
##
##  MntFishProducts  MntSweetProducts  MntGoldProds    NumDealsPurchases
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  3.00   1st Qu.:  1.00   1st Qu.:  9.00   1st Qu.: 1.000
```

```
##  Median : 12.00   Median :  8.00   Median : 24.00   Median : 2.000
##  Mean   : 37.64   Mean   : 27.03   Mean   : 43.91   Mean   : 2.325
##  3rd Qu.: 50.00   3rd Qu.: 33.00   3rd Qu.: 56.00   3rd Qu.: 3.000
##  Max.   :259.00   Max.   :262.00   Max.   :321.00   Max.   :15.000
##
##  NumWebPurchases  NumCatalogPurchases NumStorePurchases NumWebVisitsMonth
##  Min.   : 0.000   Min.   : 0.000      Min.   : 0.000    Min.   : 0.000
##  1st Qu.: 2.000   1st Qu.: 0.000      1st Qu.: 3.000    1st Qu.: 3.000
##  Median : 4.000   Median : 2.000      Median : 5.000    Median : 6.000
##  Mean   : 4.088   Mean   : 2.671      Mean   : 5.805    Mean   : 5.322
##  3rd Qu.: 6.000   3rd Qu.: 4.000      3rd Qu.: 8.000    3rd Qu.: 7.000
##  Max.   :27.000   Max.   :28.000      Max.   :13.000    Max.   :20.000
##
##   AcceptedCmp3      AcceptedCmp4      AcceptedCmp5      AcceptedCmp1
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.07366   Mean   :0.07411   Mean   :0.07275   Mean   :0.06417
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##   AcceptedCmp2        Complain           Response           age
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.0000   Min.   :18.00
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:37.00
##  Median :0.00000   Median :0.000000   Median :0.0000   Median :44.00
##  Mean   :0.01356   Mean   :0.009038   Mean   :0.1505   Mean   :45.08
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:55.00
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.0000   Max.   :74.00
##
##         relationshipStatus   educationLevel  hasChildren      numPurchase
s
##  Divorced         :231      Bachelors: 252   Min.   :0.0000   Min.   : 0.0
0
##  In a Relationship:572      Masters  :1481   1st Qu.:0.0000   1st Qu.: 6.0
0
##  Married          :857      PhD      : 480   Median :1.0000   Median :12.0
0
##  Single           :477                       Mean   :0.7144   Mean   :12.5
6
##  Widow            : 76                       3rd Qu.:1.0000   3rd Qu.:18.0
0
##                                              Max.   :1.0000   Max.   :32.0
0
##
##     amtSpent    customerFromDate     yearCust      custLength
##  Min.   :   5   Min.   :2012-07-30   2012: 490   Min.   : 1.143
##  1st Qu.:  69   1st Qu.:2013-01-16   2013:1171   1st Qu.: 7.571
##  Median : 397   Median :2013-07-08   2014: 552   Median :13.857
##  Mean   : 607   Mean   :2013-07-10               Mean   :13.776
##  3rd Qu.:1048   3rd Qu.:2013-12-31               3rd Qu.:20.036
```

```
##  Max.   :2525   Max.   :2014-06-29              Max.   :26.107
##
```

Data Preprocessing

I want to use an SVM classifier as one of my two classifiers to classify a customer as a Catalog Shopper (or not) because my data set is high-dimensional and can be easily converted to fully numeric variables. I also want to normalize my variables, setting all to a 0-1 scale (I have several binary variables already, and the dummy variables will add to that). But first, I want to reduce some of the dimensionality of my data to remove covariant variables.

I removed variables Year_Birth (because I calculated age instead), Kidshome and Teenhome (because I calculated total number of children in the home instead), customerFromDate and yearCustomer (I think custLength is best measure here), as well as the 5 AcceptedCmp variables (because Recency and NumberDealsPurchases create nice stand-ins for what those variables tell us about their campaign response rates. These can be useful variables for other analyses, but not necessarily clustering and SVM for catalog purchasing classification). This left 22 variables. After doing a quick pairwise correlation check, I decided against including the summary variables numPurchases and amtSpent because, while not covariant with variables per se, it was high enough that I didn't believe the correlation justified the added dimensionality.

I then created dummy variables for my two categorical variables, relationshipStatus and educationLevel, by using the dummyVars function from the caret package to create each dummy variable by column, then predicting the results into a new data frame. I bound the three data frames together and removed the original dummy variables relationshipStatus and educationLevel.

Last, I normalized my data using the caret library's preProcess function, using a method range with range bounds 0-1. As a final set, I created two sets ready for clustering and classification: campaignNorm, with all normalized predictor variables, and campaignClass, which is identical to campaignNorm but it includes the classification label "catPurchaser". A summary of campaignClass is below.

```
##      Income           Recency          MntWines         MntFruits
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.05040   1st Qu.:0.2424   1st Qu.:0.01607   1st Qu.:0.01005
##  Median :0.07466   Median :0.4949   Median :0.11721   Median :0.04020
##  Mean   :0.07596   Mean   :0.4950   Mean   :0.20439   Mean   :0.13228
##  3rd Qu.:0.10040   3rd Qu.:0.7475   3rd Qu.:0.33825   3rd Qu.:0.16583
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##  MntMeatProducts    MntFishProducts   MntSweetProducts    MntGoldProds
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.009275   1st Qu.:0.01158   1st Qu.:0.003817   1st Qu.:0.02804
##  Median :0.039420   Median :0.04633   Median :0.030534   Median :0.07477
##  Mean   :0.096790   Mean   :0.14531   Mean   :0.103186   Mean   :0.13680
##  3rd Qu.:0.134493   3rd Qu.:0.19305   3rd Qu.:0.125954   3rd Qu.:0.17445
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.000000   Max.   :1.00000
##  NumDealsPurchases NumWebPurchases   NumCatalogPurchases NumStorePurchases
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000     Min.   :0.0000
##  1st Qu.:0.06667   1st Qu.:0.07407   1st Qu.:0.00000     1st Qu.:0.2308
##  Median :0.13333   Median :0.14815   Median :0.07143     Median :0.3846
##  Mean   :0.15502   Mean   :0.15140   Mean   :0.09541     Mean   :0.4466
##  3rd Qu.:0.20000   3rd Qu.:0.22222   3rd Qu.:0.14286     3rd Qu.:0.6154
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000     Max.   :1.0000
```

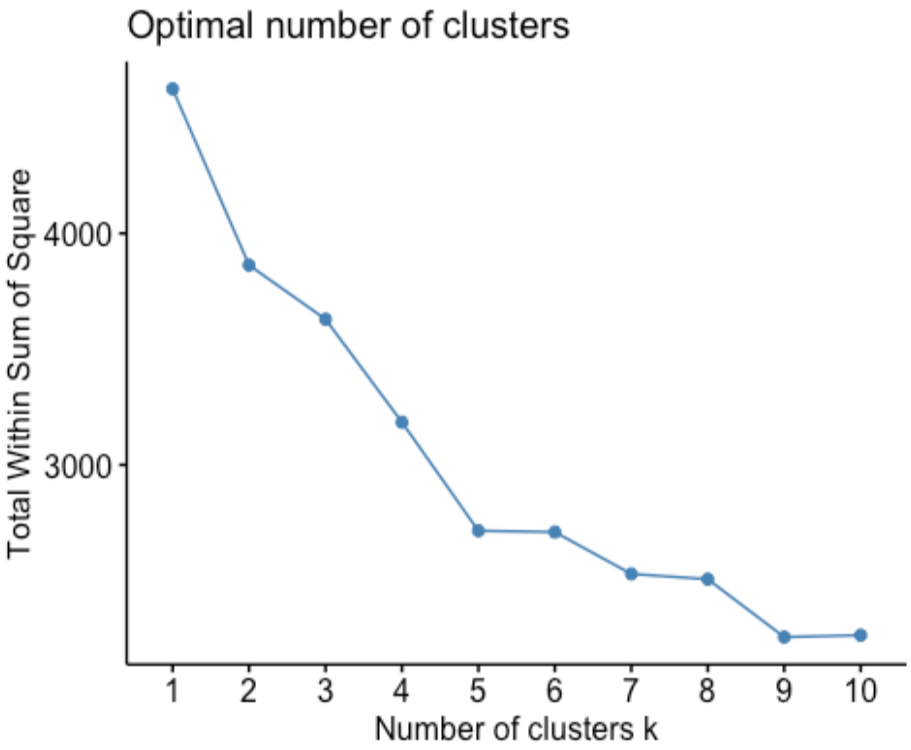
```
##  NumWebVisitsMonth    Complain           Response           age
##  Min.   :0.0000    Min.   :0.000000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.1500    1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:0.3393
##  Median :0.3000    Median :0.000000   Median :0.0000   Median :0.4643
##  Mean   :0.2661    Mean   :0.009038   Mean   :0.1505   Mean   :0.4836
##  3rd Qu.:0.3500    3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:0.6607
##  Max.   :1.0000    Max.   :1.000000   Max.   :1.0000   Max.   :1.0000
##   hasChildren       custLength     educationLevel.Bachelors
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.2575   1st Qu.:0.0000
##  Median :1.0000   Median :0.5093   Median :0.0000
##  Mean   :0.7144   Mean   :0.5061   Mean   :0.1139
##  3rd Qu.:1.0000   3rd Qu.:0.7568   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  educationLevel.Masters educationLevel.PhD relationshipStatus.Divorced
##  Min.   :0.0000         Min.   :0.0000     Min.   :0.0000
##  1st Qu.:0.0000         1st Qu.:0.0000     1st Qu.:0.0000
##  Median :1.0000         Median :0.0000     Median :0.0000
##  Mean   :0.6692         Mean   :0.2169     Mean   :0.1044
##  3rd Qu.:1.0000         3rd Qu.:0.0000     3rd Qu.:0.0000
##  Max.   :1.0000         Max.   :1.0000     Max.   :1.0000
##  relationshipStatus.In.a.Relationship relationshipStatus.Married
##  Min.   :0.0000                       Min.   :0.0000
##  1st Qu.:0.0000                       1st Qu.:0.0000
##  Median :0.0000                       Median :0.0000
##  Mean   :0.2585                       Mean   :0.3873
##  3rd Qu.:1.0000                       3rd Qu.:1.0000
##  Max.   :1.0000                       Max.   :1.0000
##  relationshipStatus.Single relationshipStatus.Widow  catPurchaser
##  Min.   :0.0000            Min.   :0.00000          Min.   :0.0000
##  1st Qu.:0.0000            1st Qu.:0.00000          1st Qu.:0.0000
##  Median :0.0000            Median :0.00000          Median :1.0000
##  Mean   :0.2155            Mean   :0.03434          Mean   :0.7402
##  3rd Qu.:0.0000            3rd Qu.:0.00000          3rd Qu.:1.0000
##  Max.   :1.0000            Max.   :1.00000          Max.   :1.0000
```

Clustering

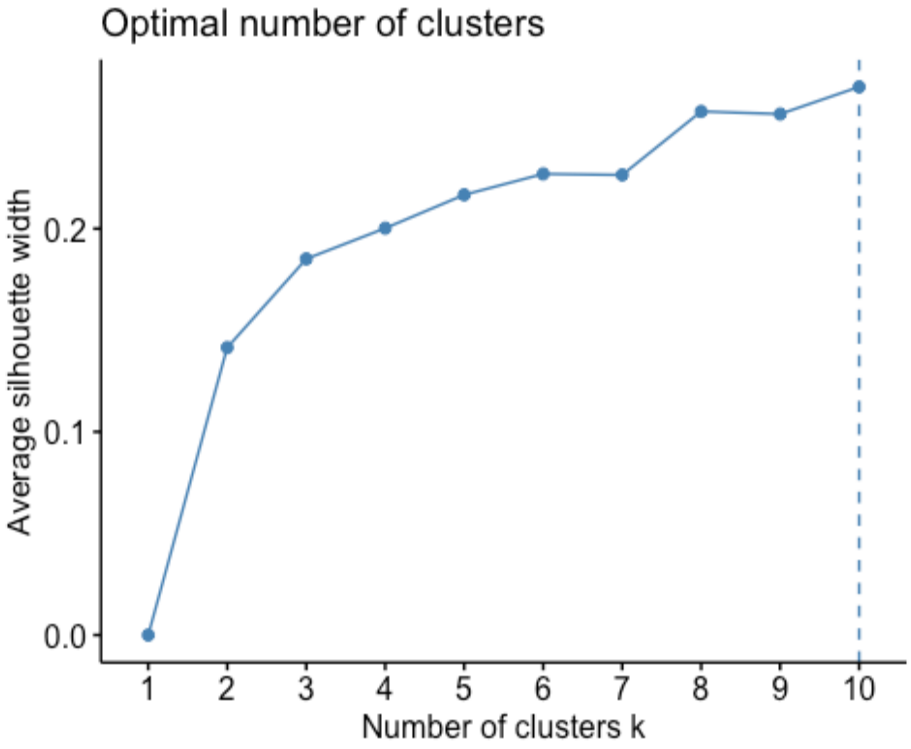
I've chosen to use k-means clustering for this data set because my data has been normalized and outliers handled, so some of the pitfalls of k-means are not factors.

I started clustering by choosing the number of clusters:

```
library(caret)
library(factoextra)

set.seed(123)
fviz_nbclust(campaignNorm, kmeans, method = "wss")
```



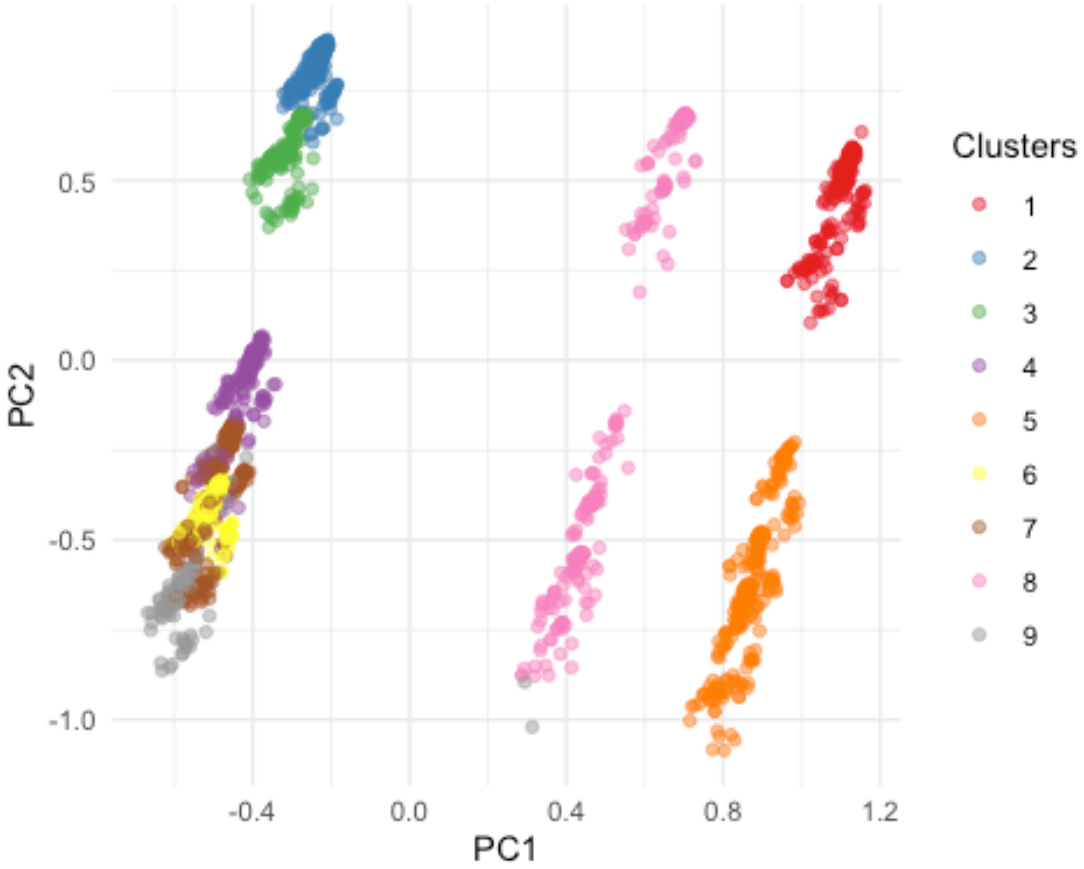

```
fviz_nbclust(campaignNorm, kmeans, method = "silhouette")
```



The scree plot showed potential clusters at 5, 7, or 9, where the silhouette showed 10. I went with 9 clusters based on the it being a strong value in both methods.

Next, I performed the clustering and visualized:

```
clusterCampaign <- kmeans(campaignNorm, centers = 9, nstart = 25)
pca <- prcomp(campaignNorm)
rotatedClusterData <- as.data.frame(pca$x)
rotatedClusterData$Clusters <- as.factor(clusterCampaign$cluster)

ggplot(data = rotatedClusterData, aes(x = PC1, y = PC2, col = Clusters)) +
  geom_point(alpha = 0.5) +
  scale_colour_brewer(palette="Set1") +
  theme_minimal()
```



It looks like of the 9 customer clusters, 6 clusters are actually quite closely related, while the other three might have separate characteristics. If I were to continue exploring this data, I might try to find some way of segmenting these three distinct patterns to overlay on my 9 clusters to provide maximum value to my marketing team.

Classification - SVM

I chose to first do an SVM Linear classifier to predict catalog purchases. Early tests of the classifier revealed that because catPurchaser was based off NumCatPurchases, the classifier was returning 100% accuracy. Since the hypothetical use case here is targeting customers and prospects for whom we may not have this information in order to determine whether to mail them a catalog, I removed

the NumPurchases columns for Store, Web, and Catalog from the campaignClass set. I also used a stratified cross-validation, since the data set is skewed towards catalog purchasers.

I first built a model without tuning:

```
folds <- 10
idx <- createFolds(campaignClass$catPurchaser, folds, returnTrain = T)
train_control_strat <- trainControl(index = idx, method = "cv", number = fold
s)

svmCampaign <- train(catPurchaser ~ ., data = campaignClass, method = "svmLin
ear",
                     trControl = train_control_strat) #pre-scaled so not incl
uding here

svmCampaign

## Support Vector Machines with Linear Kernel
##
## 2213 samples
##   26 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1992, 1992, 1992, 1992, 1992, 1991, ...
## Resampling results:
##
##   Accuracy       Kappa
##   0.9172857      0.0788956
##
## Tuning parameter 'C' was held constant at a value of 1
```

The model displayed 91.73% accuracy at predicting catalog purchasers, which suggests my team may receive a strong return on investment from sending catalogs to potential customers.

I also ran the model to tune the C:

```
grid <- expand.grid(C = 10^seq(-5,2,0.5))
svm_grid <- train(catPurchaser ~ ., data = campaignClass, method = "svmLinear
",
                  trControl = train_control_strat, tuneGrid = grid)

svm_grid

## Support Vector Machines with Linear Kernel
##
## 2213 samples
##   26 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1992, 1992, 1992, 1992, 1992, 1991, ...
```

```
## Resampling results across tuning parameters:
##
##   C             Accuracy   Kappa
##   1.000000e-05  0.7400574  0.0000000
##   3.162278e-05  0.7400574  0.0000000
##   1.000000e-04  0.7400574  0.0000000
##   3.162278e-04  0.7400574  0.0000000
##   1.000000e-03  0.8114737  0.3775397
##   3.162278e-03  0.8933342  0.7362555
##   1.000000e-02  0.9010164  0.7566723
##   3.162278e-02  0.9095974  0.7741348
##   1.000000e-01  0.9195563  0.7967278
##   3.162278e-01  0.9209036  0.7994498
##   1.000000e+00  0.9172857  0.7889560
##   3.162278e+00  0.9172734  0.7889711
##   1.000000e+01  0.9199761  0.7958865
##   3.162278e+01  0.9181743  0.7905303
##   1.000000e+02  0.9181743  0.7903302
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 1.
```

Cs from 0.03 to 100 all had high accuracy scores, so I'd likely continue using C = 1 if further revising the model.

Classification - Decision Tree

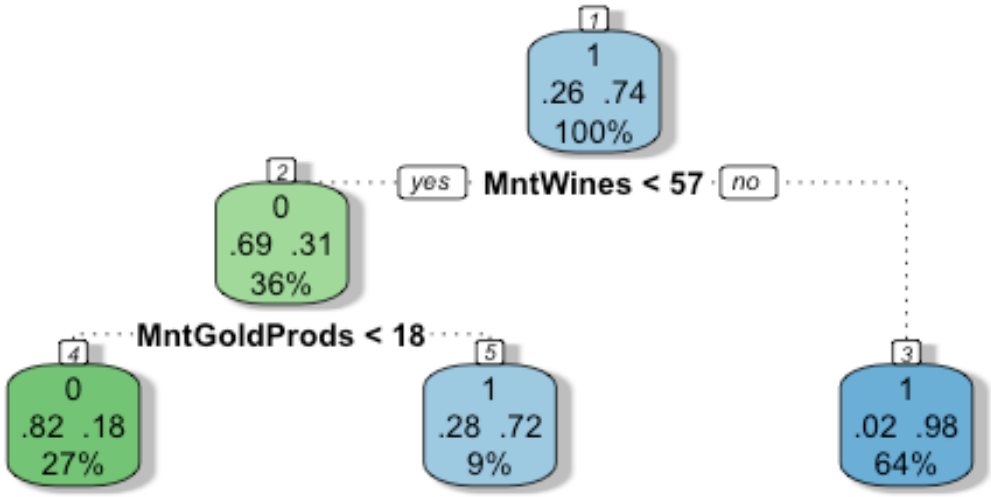
I also classified via decision tree. I do not believe this method will be better than SVM because there are too many variables. To model the tree, I used a non-normalized, non-dummified version of the data set, since those aren't required for a decision tree and I thought the dummies in particular might add too many unneeded variables for a tree to process.

I started with an un-tuned tree:

```
library(rpart)
library(rattle)

treeCampaign2 <- train(catPurchaser ~ ., data = campaignTree, method = "rpart
1SE",
                       trControl = train_control_strat)
treeCampaign2
```

```
## CART
##
## 2213 samples
##   17 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1992, 1992, 1992, 1992, 1992, 1991, ...
## Resampling results:
```

```
##
##   Accuracy   Kappa
##   0.9051119  0.7531281

fancyRpartPlot(treeCampaign2$finalModel, caption = "")
```



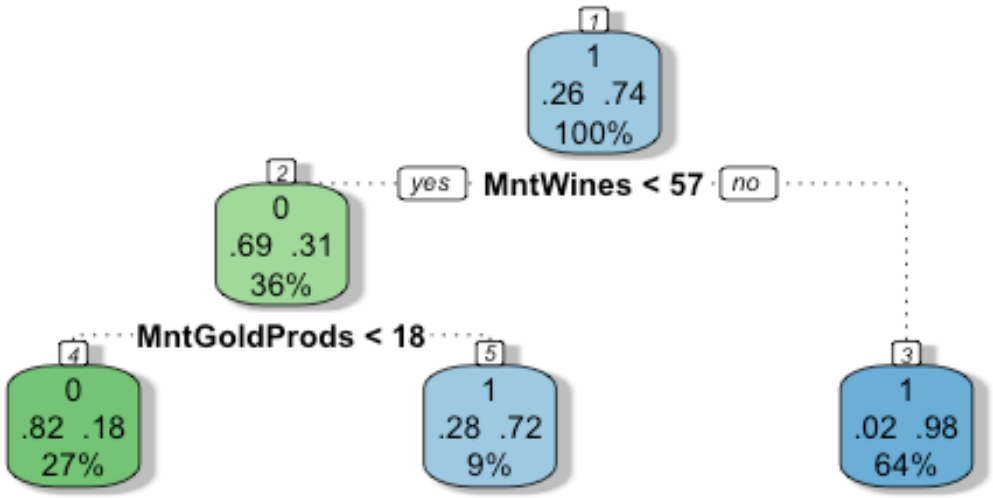
My tree has three levels, and it looks (without a confusion matrix) to be roughly accurate. The tree suggests that buyers with low purchase values in gold and/or wine are not catalog purchasers.

I also tried tuning the minsplit and minbucket parameters. Because the default depth is 30 and I have a lot of variables, I didn't want to unnecessarily limit the model from selecting among variables.

```
hypers = rpart.control(minsplit = 200)
treeCampaign3 <- train(catPurchaser ~ ., data = campaignTree, control = hyper
s,
                       trControl = train_control_strat, method = "rpart1SE")
treeCampaign3
```

```
## CART
##
## 2213 samples
##   17 predictor
##    2 classes: '0', '1'
##
## No pre-processing
```

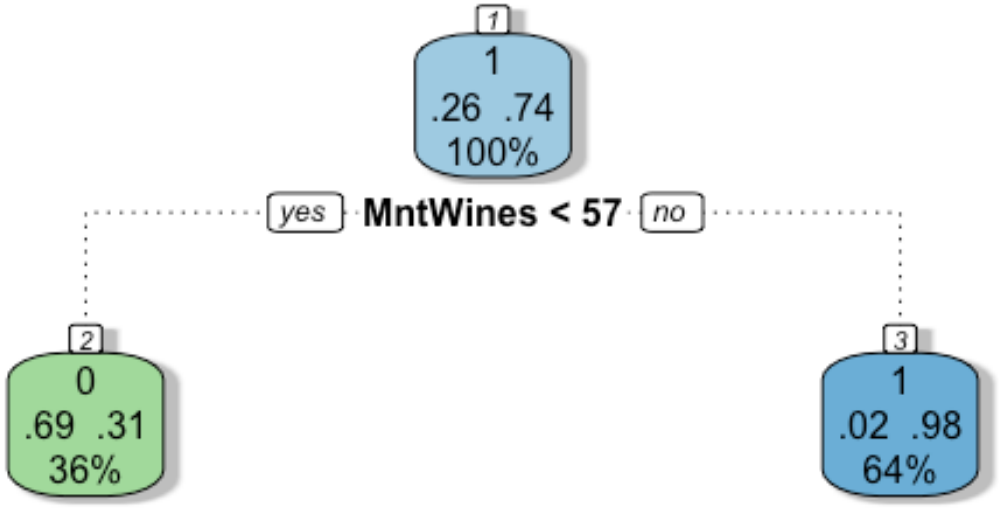
```
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1992, 1992, 1992, 1992, 1992, 1991, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9051119  0.7524733

fancyRpartPlot(treeCampaign3$finalModel, caption = "")
```



Tuning minsplit alone creates the same model.

```
hypers = rpart.control(minsplit = 400, minbucket = 300)
treeCampaign4 <- train(catPurchaser ~ ., data = campaignTree, control = hyper
s,
                       trControl = train_control_strat, method = "rpart1SE")
treeCampaign4

## CART
##
## 2213 samples
##   17 predictor
##    2 classes: '0', '1'
##
## No pre-processing
```

```
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1992, 1992, 1992, 1992, 1992, 1991, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.859488  0.6342188

fancyRpartPlot(treeCampaign4$finalModel, caption = "")
```



Tuning minsplit and minbucket oversimplifies the model. For evaluation purposes, I'll choose between the base SVM model and tree2 (original tree).

The two classification models performed remarkably similarly on a confusion matrix evaluation[1]. Because the cost of sending a catalog is non-zero, I'll favor the classification model that has a slightly lower rate of false positives, the SVM classifier.

[1] RMarkdown was experiencing difficulties recreating this matrix for reasons unknown, so recreating a simplified version of the matrix below.

CONFUSION MATRIX, SVM CLASSIFIER	REFERENCE 0	REFERENCE 1
PREDICTION 0	507	68
PREDICTION 1	100	1537

CONFUSION MATRIX, TREE CLASSIFIER	REFERENCE 0	REFERENCE 1
PREDICTION 0	489	86
PREDICTION 1	108	1529

Evaluation

To evaluate my classifier, I'll use the SVM confusion matrix above.

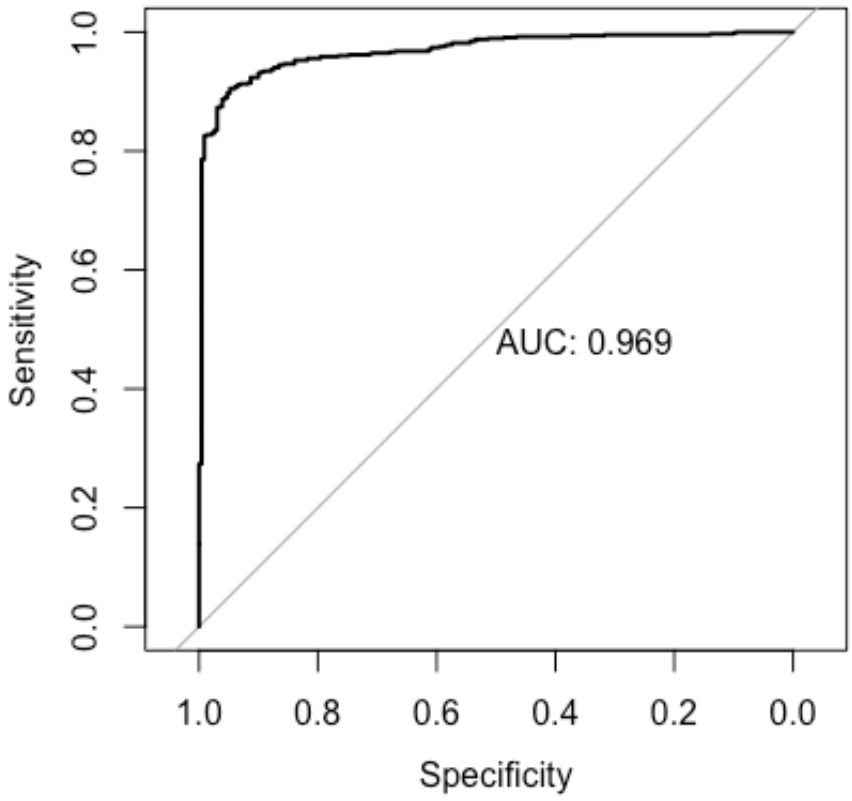
I calculated recall and precision rates (confirmed with the confusion matrix's metrics) as follows. Both recall and precision are quite high on this classifier:

```
svmPrecision <- 507/(507 + 68)
svmRecall <- 507/(507 + 100)
svmRecall

## [1] 0.8352554

svmPrecision

## [1] 0.8817391
```

I then calculated the ROC.

```
library(pROC)

index <- createDataPartition(y = campaignROC$catPurchaser2, p = 0.6, list = F
ALSE)
trainCamp <- campaignROC[index,]
testCamp <- campaignROC[-index,]

train_control <- trainControl(method = "cv", number = folds, classProbs = TRU
E)
```

```
svmCampaign2 <- train(catPurchaser2 ~ ., data = trainCamp, method = "svmLinea
r",
                      trControl = train_control)

pred_svm_prob <- predict(svmCampaign2, testCamp, type = "prob")
roc_obj <- roc((testCamp$catPurchaser2),pred_svm_prob[,1])

plot(roc_obj, print.auc = TRUE)
```



The area under the curve for the SVM classifier is high. This appears to be a good way of predicting if a customer will purchase from the catalog.

Report Conclusions

I specifically chose this data set because I wanted to work with a big set of data with some error and mess, particularly with a lot of variables to see how the concepts we applied in this class would scale with unknown, highly dimensional data. I think the biggest takeaway from the data was that the clustering in particular is not the end of finding insights in data. The 9 customer clusters I discovered have utility beyond just understanding their commonalities, and the visualizations helped outline a path for further development. In addition, it was interesting to see how well a somewhat skewed classification label performed in an SVM model, and it would be interesting to test this model out on further generated data to see how it performs on unseen data.

Alexandra DeGrandchamp
November 19, 2022

Reflection

I so enjoyed my quarter in this course. As a data analyst and data manager, I've been feeling burnt out and lost in my profession. After a tough year professionally and an elongated job search, I had been feeling like I couldn't really "data" like other people "data". This course has given me new purpose and skills that I've already put to use in my professional life. For example, my team builds decision trees - a bit manually, to be sure - for client data for manual classification purposes to complex taxonomies. Understanding the foundation and theory behind decision trees, and building some from sample data sets, gave me a new appreciation and way of approaching client data that I was lacking previously. I had been considering leaving analytics altogether for the data engineering space, but I am eager to give analytics, and maybe eventually an ML or AI role, another go after this course!