Clustering

Eoin Flynn

26 March 2018

Bond University
Data Science
Final Assignment

Contents

Introduction

In this report we will using an unsupervised machine learning technique called clustering to identify groups of customers within our dataset. The model will produce a set of clusters and return a "typical customer" for those groupings. A typical customer is the type of person we expect to see in that group, for example a typical customer who uses tech support might be a senior who pays month-to-month. Being able to identify groups of customers will allow you to make more informed decisions when marketing new products or entering new markets.

Functions

This section will hold all of the functions that will be used throughout this markdown.

```r
# Change rows to factors
setRowAsFactor <- function(dataset, columns) {
    for (column in columns) {
        dataset[, column] <- as.factor(dataset[, column])
    }
    return(dataset)
}

# Create a new customer for predicition. Returns a dataframe
createCustomer <- function(originalDataset, gender, SeniorCitizen,
    Partner, Dependents, tenure, PhoneService, MultipleLines,
    InternetService, OnlineSecurity, OnlineBackup, DeviceProtection,
    TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling,
    PaymentMethod, MontlyCharges, TotalCharges, Churn) {
    # Create a copy of the original dataset and keep one row that
    # will be overridden with the new data.
    newCustomer <- customerDataset[1, ]

    newCustomer$gender <- gender
    newCustomer$SeniorCitizen <- SeniorCitizen
    newCustomer$Partner <- Partner
    newCustomer$Dependents <- Dependents
    newCustomer$tenure <- tenure
    newCustomer$PhoneService <- PhoneService
    newCustomer$MultipleLines <- MultipleLines
    newCustomer$InternetService <- InternetService
    newCustomer$OnlineSecurity <- OnlineSecurity
    newCustomer$OnlineBackup <- OnlineBackup
    newCustomer$DeviceProtection <- DeviceProtection
    newCustomer$TechSupport <- TechSupport
    newCustomer$StreamingTV <- StreamingTV
    newCustomer$StreamingMovies <- StreamingMovies
    newCustomer$Contract <- Contract
    newCustomer$PaperlessBilling <- PaperlessBilling
    newCustomer$PaymentMethod <- PaymentMethod
    newCustomer$MonthlyCharges <- MontlyCharges
    newCustomer$TotalCharges <- TotalCharges
    newCustomer$Churn <- Churn

    # Convert fields that are factors
```

```
    newCustomer <- setRowAsFactor(newCustomer, c("gender", "SeniorCitizen",
        "Partner", "Dependents", "PhoneService", "MultipleLines",
        "InternetService", "OnlineSecurity", "OnlineBackup",
        "DeviceProtection", "TechSupport", "StreamingTV", "StreamingMovies",
        "Contract", "PaperlessBilling", "PaymentMethod", "Churn"))

    return(newCustomer)
}

# Gets a dataframe from a locally hosted MySQL server.
# Returns a dataframe
loadDataframeFromMySQL <- function(user, password, host = "localhost",
    dbname, statement, port = 3306) {
    suppressMessages(library(RMySQL))

    # Connect to the server
    dataBase <- dbConnect(MySQL(), user = user, password = password,
        host = host, dbname = dbname, port = port)
    # Retrieve the info the from the specified server
    dataframe <- dbGetQuery(dataBase, statement = statement)
    # Close the connection to the server
    dbDisconnect(dataBase)

    return(dataframe)

}
```

Data

In this section we will load in our data and do some basic data exploration.

```
customerDataset <- loadDataframeFromMySQL(user = "root", password = "A13337995",
    dbname = "world", statement = "Select * from world.customerChurn")

# Loop through and change all relevant rows to factors and
# returns the dataset post modification customerDataset <-
# setRowAsFactor(customerDataset, c('gender',
# 'SeniorCitizen', 'Partner', 'Dependents', 'PhoneService',
# 'MultipleLines', 'InternetService', 'OnlineSecurity',
# 'OnlineBackup', 'DeviceProtection','TechSupport',
# 'StreamingTV', 'StreamingMovies', 'Contract',
# 'PaperlessBilling', 'PaymentMethod', 'Churn' ))

# Drop the columns that will not be needed
customerDataset <- customerDataset[, -which(names(customerDataset) %in%
    c("customerID", "MultipleLines", "OnlineSecurity", "OnlineBackup",
        "DeviceProtection", "TechSupport", "StreamingTV", "StreamingMovies",
        "PaymentMethod"))]

customerDataset$gender[customerDataset$gender == "Female"] <- 1
customerDataset$gender[customerDataset$gender == "Male"] <- 0
```

```r
customerDataset$Partner[customerDataset$Partner == "Yes"] <- 1
customerDataset$Partner[customerDataset$Partner == "No"] <- 0

customerDataset$Dependents[customerDataset$Dependents == "Yes"] <- 1
customerDataset$Dependents[customerDataset$Dependents == "No"] <- 0

customerDataset$PhoneService[customerDataset$PhoneService ==
    "Yes"] <- 1
customerDataset$PhoneService[customerDataset$PhoneService ==
    "No"] <- 0

customerDataset$PaperlessBilling[customerDataset$PaperlessBilling ==
    "Yes"] <- 1
customerDataset$PaperlessBilling[customerDataset$PaperlessBilling ==
    "No"] <- 0

# 1 if a customer has internet, 0 if not
customerDataset$InternetService[customerDataset$InternetService ==
    "Fiber optic"] <- 1
customerDataset$InternetService[customerDataset$InternetService ==
    "DSL"] <- 1
customerDataset$InternetService[customerDataset$InternetService ==
    "No"] <- 0

# 1 if a customer is not on a yearly or bi-yearly contract
# (not locked in)
customerDataset$Contract[customerDataset$Contract == "Month-to-month"] <- 1
customerDataset$Contract[customerDataset$Contract == "One year"] <- 0
customerDataset$Contract[customerDataset$Contract == "Two year"] <- 0

customerDataset$Churn[customerDataset$Churn == "Yes"] <- 1
customerDataset$Churn[customerDataset$Churn == "No"] <- 0

customerDataset <- customerDataset[complete.cases(customerDataset),
    ]
```

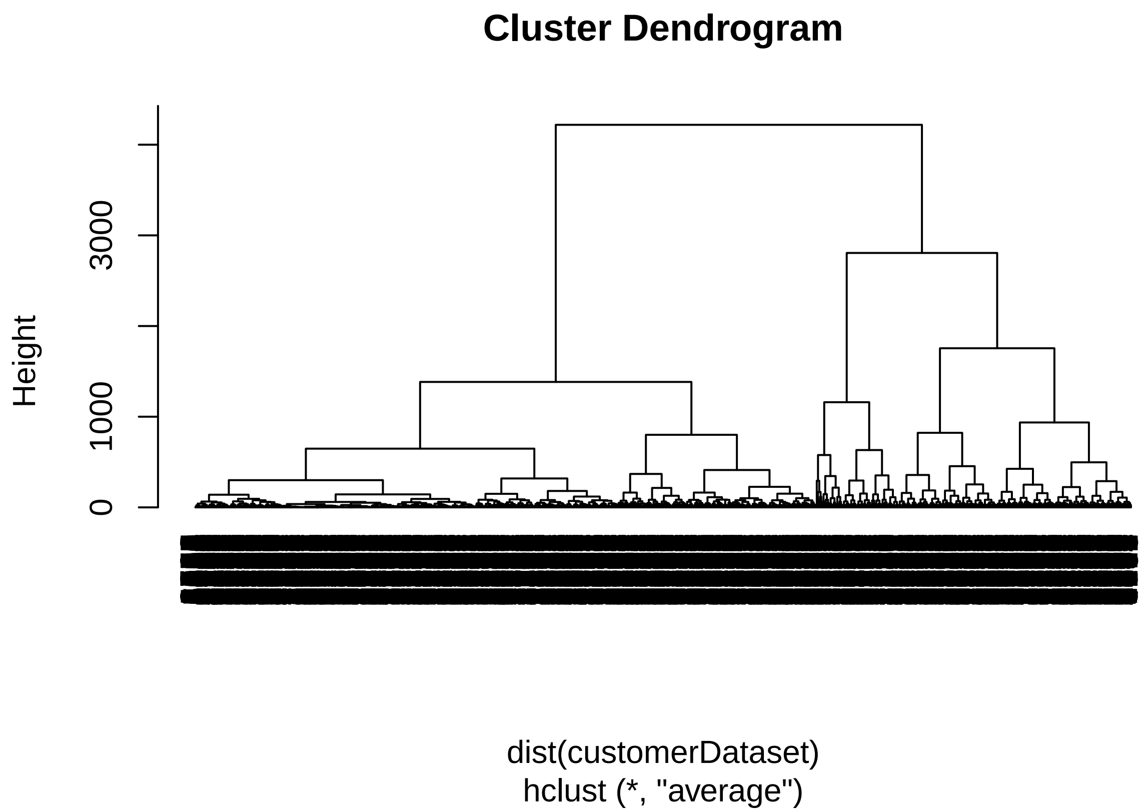
Model

To determine the optimal number of clusters we will first create a dendogram view how far we can drill down and then produce a series of models using different values that look reasonable on the dendogram. Picking the right number of clusters is highly subjective and varies by dataset so there is no golden number, so that is why we are creating our series of models and presenting the one to management which has the best insight.

Dendogram

To determine the optimal number of clusters we will first create a dendogram view how far we can drill down and then produce a series of models using different values that look reasonable on the dendogram. Picking the right number of clusters is highly subjective and varies by dataset so there is no golden number, so that is why we are creating our series of models and presenting the one to management which has the best insight.

```r
hierarchicalClustering <- hclust(dist(customerDataset), method = "ave")
plot(hierarchicalClustering, hang = -1)
```



###Dendogram Discussion
We can see from the dendogram plot that there are so many groupings that it becomes a blur where we cannot make-out any groupings at all. If we were to create a clustering model which drills down all the way then it would have zero insight for management since it would apply to such a finite grouping of customers, on the flip side, if we use a model with too few customers then the model will lack specificity and thus also provide little to no insight to management. Looking at the dendogram we can see that between 14 and 17 clusters breaks the data down so that it is not too specific, but also not too general

Cluster Models

Based off our observations from the dendogram, we will now create a series of models and compare them to analyse which has the greatest insight for managment.

```r
set.seed(12216)
fourteenClusterModel <- kmeans(customerDataset, 14)
fiveClusterModel <- kmeans(customerDataset, 15)
sixClusterModel <- kmeans(customerDataset, 16)
sevenClusterModel <- kmeans(customerDataset, 17)

fourteenClusterModel$centers
```

```
##       gender SeniorCitizen   Partner Dependents    tenure PhoneService
## 1  0.4863760     0.1205722 0.2316076  0.2152589  3.332425    0.8876022
```

```
## 2  0.5172811     0.1209677 0.3721198  0.2776498 12.975806    0.8928571
## 3  0.4921136     0.2176656 0.5835962  0.3249211 52.533123    0.9085174
## 4  0.4353234     0.2039801 0.5074627  0.2412935 34.718905    0.8432836
## 5  0.4896907     0.2061856 0.8092784  0.3608247 71.134021    1.0000000
## 6  0.5291829     0.1614786 0.5428016  0.2957198 37.715953    0.8696498
## 7  0.5227273     0.1915584 0.5974026  0.3181818 47.753247    0.8831169
## 8  0.5019763     0.2569170 0.7549407  0.3438735 67.664032    1.0000000
## 9  0.4897959     0.2419825 0.5422741  0.3177843 42.416910    0.8221574
## 10 0.4608696     0.1405797 0.5246377  0.3623188 36.563768    0.9000000
## 11 0.4808260     0.1887906 0.6666667  0.3510324 56.569322    0.9410029
## 12 0.5206612     0.1225895 0.4214876  0.3526171 25.596419    0.8939394
## 13 0.5051195     0.1911263 0.7645051  0.3174061 65.464164    1.0000000
## 14 0.5015773     0.2302839 0.7129338  0.3406940 61.492114    1.0000000
##    InternetService   Contract PaperlessBilling MonthlyCharges TotalCharges
## 1        0.6294278 0.89441417        0.5115804       45.68730     115.7797
## 2        0.6463134 0.71198157        0.5472350       50.55547     465.3058
## 3        1.0000000 0.34700315        0.7034700       82.31278    4175.7388
## 4        1.0000000 0.61194030        0.6791045       74.84876    2407.6704
## 5        1.0000000 0.02061856        0.7783505      111.60335    7955.4892
## 6        0.7665370 0.53501946        0.5525292       61.24018    1827.0456
## 7        1.0000000 0.43506494        0.6590909       78.99919    3586.1180
## 8        1.0000000 0.13833992        0.7747036      104.15455    7041.3223
## 9        1.0000000 0.51895044        0.6413994       75.79257    3005.0894
## 10       0.6173913 0.48695652        0.5217391       51.70725    1329.4068
## 11       1.0000000 0.31563422        0.6843658       87.69646    4832.2419
## 12       0.6074380 0.54407713        0.5371901       49.75634     883.1537
## 13       1.0000000 0.17747440        0.6825939       96.02509    6255.3261
## 14       1.0000000 0.22712934        0.6624606       91.65237    5558.7830
##         Churn
## 1  0.45435967
## 2  0.29608295
## 3  0.16719243
## 4  0.29601990
## 5  0.08247423
## 6  0.18287938
## 7  0.16558442
## 8  0.13438735
## 9  0.26530612
## 10 0.21884058
## 11 0.16814159
## 12 0.25895317
## 13 0.14675768
## 14 0.15141956
```

```
fiveClusterModel$centers
```

```
##       gender SeniorCitizen   Partner Dependents    tenure PhoneService
## 1  0.4914773     0.2386364 0.5454545  0.3125000 42.761364    0.8238636
## 2  0.5102041     0.2478134 0.7580175  0.3352770 66.725948    1.0000000
## 3  0.5167464     0.1722488 0.5406699  0.2846890 37.210526    0.8755981
## 4  0.4830876     0.1231570 0.2098873  0.1968777  2.314831    0.8837814
## 5  0.5070423     0.1161972 0.3961268  0.2852113 15.855634    0.9014085
## 6  0.4455696     0.2050633 0.5088608  0.2481013 35.207595    0.8405063
## 7  0.4615385     0.2105263 0.7975709  0.3643725 70.728745    1.0000000
## 8  0.5322997     0.1963824 0.7416021  0.3255814 62.987080    1.0000000
```

```
## 9  0.5131965     0.1187683 0.3255132  0.2785924  8.802053    0.8885630
## 10 0.5000000     0.1259690 0.4224806  0.3449612 24.337209    0.8934109
## 11 0.4597701     0.2040230 0.6724138  0.3362069 57.954023    0.9655172
## 12 0.4922049     0.1447661 0.5456570  0.3496659 38.574610    0.8797327
## 13 0.5164179     0.2029851 0.6029851  0.3223881 48.773134    0.8805970
## 14 0.4980392     0.1313725 0.4803922  0.3705882 33.690196    0.9078431
## 15 0.4924012     0.2036474 0.5987842  0.3434650 53.188450    0.9118541
##    InternetService   Contract PaperlessBilling MonthlyCharges TotalCharges
## 1        1.0000000 0.49431818        0.6363636       76.23594   3052.87259
## 2        1.0000000 0.15743440        0.7172012      101.13207   6740.64213
## 3        0.8038278 0.57655502        0.5693780       63.23517   1906.16758
## 4        0.6305291 0.92714657        0.5125759       45.08664     82.00278
## 5        0.6355634 0.66725352        0.5422535       50.22606    562.00863
## 6        1.0000000 0.58987342        0.6784810       74.83405   2445.98886
## 7        1.0000000 0.02834008        0.7854251      110.59696   7833.08320
## 8        1.0000000 0.22480620        0.6744186       93.19806   5813.67481
## 9        0.6304985 0.75219941        0.5205279       48.72669    302.56224
## 10       0.6298450 0.56976744        0.5484496       50.80930    859.43983
## 11       1.0000000 0.29022989        0.6839080       88.19713   4991.96695
## 12       0.6547884 0.47884187        0.5322940       53.94477   1503.79154
## 13       1.0000000 0.42089552        0.6716418       79.17672   3671.83582
## 14       0.5941176 0.49803922        0.5254902       50.73520   1177.06373
## 15       1.0000000 0.34346505        0.6990881       83.54210   4302.57462
##         Churn
## 1  0.25284091
## 2  0.13702624
## 3  0.20574163
## 4  0.48829141
## 5  0.26936620
## 6  0.29113924
## 7  0.09716599
## 8  0.14987080
## 9  0.32404692
## 10 0.26162791
## 11 0.15804598
## 12 0.19376392
## 13 0.15820896
## 14 0.24117647
## 15 0.18237082
```

```
sixClusterModel$centers
```

```
##       gender SeniorCitizen   Partner Dependents    tenure PhoneService
## 1  0.4447301     0.2005141 0.5089974  0.2442159 35.089974    0.8431877
## 2  0.4980392     0.1313725 0.4803922  0.3705882 33.690196    0.9078431
## 3  0.4830876     0.1231570 0.2098873  0.1968777  2.314831    0.8837814
## 4  0.4834835     0.2192192 0.5975976  0.3363363 53.153153    0.9039039
## 5  0.5167464     0.1722488 0.5406699  0.2846890 37.210526    0.8755981
## 6  0.4981949     0.1949458 0.7725632  0.3249097 65.703971    1.0000000
## 7  0.5140187     0.1931464 0.5887850  0.3115265 47.993769    0.8816199
## 8  0.4896907     0.2061856 0.8092784  0.3608247 71.134021    1.0000000
## 9  0.5131965     0.1187683 0.3255132  0.2785924  8.802053    0.8885630
## 10 0.5032468     0.2175325 0.7207792  0.3311688 61.626623    1.0000000
## 11 0.4922049     0.1447661 0.5456570  0.3496659 38.574610    0.8797327
## 12 0.5070423     0.1161972 0.3961268  0.2852113 15.855634    0.9014085
```

```
## 13 0.5019920     0.2589641 0.7529880  0.3426295 67.661355    1.0000000
## 14 0.5000000     0.1259690 0.4224806  0.3449612 24.337209    0.8934109
## 15 0.4922601     0.1919505 0.6656347  0.3467492 57.126935    0.9566563
## 16 0.4911765     0.2441176 0.5500000  0.3235294 42.723529    0.8205882
##    InternetService   Contract PaperlessBilling MonthlyCharges TotalCharges
## 1        1.0000000 0.58868895        0.6760925       74.91967   2441.38380
## 2        0.5941176 0.49803922        0.5254902       50.73520   1177.06373
## 3        0.6305291 0.92714657        0.5125759       45.08664     82.00278
## 4        1.0000000 0.35735736        0.7117117       82.60526   4243.91742
## 5        0.8038278 0.57655502        0.5693780       63.23517   1906.16758
## 6        1.0000000 0.15884477        0.6714801       95.99801   6279.56011
## 7        1.0000000 0.41433022        0.6604361       79.43224   3624.81340
## 8        1.0000000 0.02061856        0.7783505      111.60335   7955.48918
## 9        0.6304985 0.75219941        0.5205279       48.72669    302.56224
## 10       1.0000000 0.24675325        0.6655844       92.25406   5609.41575
## 11       0.6547884 0.47884187        0.5322940       53.94477   1503.79154
## 12       0.6355634 0.66725352        0.5422535       50.22606    562.00863
## 13       1.0000000 0.13944223        0.7768924      104.21394   7044.41434
## 14       0.6298450 0.56976744        0.5484496       50.80930    859.43983
## 15       1.0000000 0.29721362        0.6873065       87.95697   4904.99458
## 16       1.0000000 0.51176471        0.6323529       75.87191   3031.89706
##         Churn
## 1  0.29048843
## 2  0.24117647
## 3  0.48829141
## 4  0.18618619
## 5  0.20574163
## 6  0.14079422
## 7  0.15576324
## 8  0.08247423
## 9  0.32404692
## 10 0.16233766
## 11 0.19376392
## 12 0.26936620
## 13 0.13545817
## 14 0.26162791
## 15 0.14860681
## 16 0.26176471
```

```r
sevenClusterModel$centers
```

```
##       gender SeniorCitizen   Partner Dependents    tenure PhoneService
## 1  0.5051125     0.1329243 0.4601227  0.3701431 32.869121    0.9141104
## 2  0.5018051     0.1191336 0.3971119  0.2870036 15.714801    0.9007220
## 3  0.4981949     0.1949458 0.7725632  0.3249097 65.703971    1.0000000
## 4  0.4823151     0.1897106 0.6720257  0.3536977 57.276527    0.9581994
## 5  0.5445205     0.2157534 0.5753425  0.3219178 49.304795    0.8835616
## 6  0.5236908     0.1546135 0.5561097  0.3067332 38.623441    0.8852868
## 7  0.4830876     0.1231570 0.2098873  0.1968777  2.314831    0.8837814
## 8  0.4646465     0.2121212 0.5824916  0.3232323 44.360269    0.8383838
## 9  0.5154639     0.1178203 0.3254786  0.2783505  8.779087    0.8895434
## 10 0.5080321     0.1265060 0.4236948  0.3413655 23.596386    0.8935743
## 11 0.4539474     0.2105263 0.4967105  0.2269737 33.351974    0.8388158
## 12 0.5032468     0.2175325 0.7207792  0.3311688 61.626623    1.0000000
## 13 0.4802632     0.2039474 0.5953947  0.3322368 53.223684    0.9078947
```

```
## 14 0.4896907     0.2061856 0.8092784  0.3608247 71.134021    1.0000000
## 15 0.4753363     0.1434978 0.5627803  0.3475336 38.479821    0.8721973
## 16 0.5019920     0.2589641 0.7529880  0.3426295 67.661355    1.0000000
## 17 0.4781022     0.2299270 0.4963504  0.2810219 38.364964    0.8321168
##    InternetService   Contract PaperlessBilling MonthlyCharges TotalCharges
## 1        0.5889571 0.50511247        0.5357873       50.83200   1148.21830
## 2        0.6371841 0.66606498        0.5397112       50.16182    556.91751
## 3        1.0000000 0.15884477        0.6714801       95.99801   6279.56011
## 4        1.0000000 0.29903537        0.6816720       87.87653   4917.20000
## 5        1.0000000 0.40068493        0.6678082       79.45599   3728.63904
## 6        0.7406484 0.50623441        0.5436409       60.40636   1833.14115
## 7        0.6305291 0.92714657        0.5125759       45.08664     82.00278
## 8        1.0000000 0.49158249        0.6700337       76.47660   3189.04192
## 9        0.6303387 0.75257732        0.5228277       48.79330    301.99507
## 10       0.6385542 0.58032129        0.5582329       51.20382    843.83203
## 11       1.0000000 0.65460526        0.6875000       73.94474   2270.21250
## 12       1.0000000 0.24675325        0.6655844       92.25406   5609.41575
## 13       1.0000000 0.34539474        0.7072368       83.37812   4294.68421
## 14       1.0000000 0.02061856        0.7783505      111.60335   7955.48918
## 15       0.6479821 0.47757848        0.5044843       52.87478   1458.60684
## 16       1.0000000 0.13944223        0.7768924      104.21394   7044.41434
## 17       1.0000000 0.56569343        0.6313869       75.58193   2704.96442
##         Churn
## 1  0.24335378
## 2  0.27075812
## 3  0.14079422
## 4  0.14790997
## 5  0.15753425
## 6  0.16957606
## 7  0.48829141
## 8  0.22895623
## 9  0.32400589
## 10 0.27108434
## 11 0.29605263
## 12 0.16233766
## 13 0.19078947
## 14 0.08247423
## 15 0.20403587
## 16 0.13545817
## 17 0.27737226
```

Cluster Models Discussion

We will present the 14 cluster model to managment as it has a number of rows which will be of a great benefit to them for gaining better insight into their customers. If we look at row 14 in particular we can see a clear grouping of customers where very few churn. The group is split almost 50/50 between men and women, with slighlty more men in the group than women. The typical member of this group is very unlikely a senior but is extremely likely to have a partner. The group is always one of few where the average customer definitely has a phone service and internet service, and this may be the key as to why they are extremely unlikely to churn. Another key may be the fact that they are more likely to be a lock in contract (one or two years) as opposed being on a month to month plan. The recommendations for how this data could be used in a business sense can be found in the management section of the report.