DESCO – Trabalho exploratório de dados de clientes e compras de uma rede de perfumarias

Daniel Coelho
Instituto Superior de Engenharia do Porto
1160943@isep.ipp.pt

Ricardo Sousa
Instituto Superior de Engenharia do Porto
1160900@isep.ipp.pt

1. Introdução

No âmbito da área disciplinar de Descoberta de Conhecimento, foi requerido que se fizesse um estudo de dados fornecidos por uma rede de perfumarias, de forma a prever padrões de consumo dos seus clientes. Para tal, com o crescimento da popularidade das técnicas de análise de dados, foi requerido que utilizássemos algumas destas técnicas, após uma análise crítica das mesmas.

Este documento apresenta então a documentação literária deste estudo. O dataset fornecido contêm dois ficheiros em formato csv. O primeiro contém dados dos clientes e o segundo os dados das compras dos mesmos. Os data frames resultantes destes datasets podem ser fundidos num só através do identificador do cliente.

2. Preparação dos dados

Para ser possível utilizar os datasets para conseguir analisar e recolher informações dos dados, é necessário tratar os mesmos. Esta secção apresenta o conjunto de operações que foram efetuadas para que tal aconteça.

Em primeiro lugar, foram necessário tratar todos os dados que nos pareceram pouco congruentes. Sendo assim, corrigiu-se valores vazios, de forma que todos estes tenham a mesma estrutura (“NA”) e ajustaram-se as datas (tanto a data de registo, como a data da venda), de forma que as horas minutos e segundos não fossem apresentados (estes eram sempre 00:00:00).

```
# Vazios do MaritalStatus
emptyPosition <- which(levels(dataClients$MaritalStatus) == "" )
levels(dataClients$MaritalStatus)[emptyPosition] <- "NA"

# Retirar Horas do timestamp
dataClients$RegistrationDt <- as.Date(dataClients$RegistrationDt)
dataClients$RegistrationDt <- format(dataClients$RegistrationDt, "%d-%m-%Y")
```

Figura 1 – Preparação dados Vazios e Datas

2.1 Dataset Clients

Em específico para o dataset Clients, foi também necessário ajustar as zonas e as cidades dos clientes.

A preparação dos dados relativa ao atributo das cidades dos clientes possuiu várias etapas, entre as quais:

- Remoção de espaços antes e depois do nome das cidades;
- Alteração dos nomes para maiúsculas;
- Alteração de letras acentuadas para letras não acentuadas ou respetivos equivalentes;
- Alteração de códigos ou símbolos para respetivas letras equivalentes;
- Remoção de valores entre parênteses através do uso de regex;
- Alterações de acrónimos (St., V.N., etc) para os seus respetivos nomes;
- Remoção de acrónimos relativos a nomes de municípios ou localidades.

Esta preparação fez com que se passasse de 964 para apenas 828 cidades única no dataset.

2.2 Dataset Purchases

Já para o dataset Purchases foram executadas:

- Alterações no atributo TAX de forma que os valores fossem igualmente capitalizados;
- Alterações no atributo DESCR para que não existissem carateres especiais (no caso do dataset os “*”).

3. Pré-Processamento dos dados

Assim como na preparação dos dados, o pré-processamento foi também ele dividido entre os dois datasets.

Após o pré-processamento, os dataframes foram fundidos num só através do identificador do cliente.

3.1 Dataset Clients

No que diz respeito aos dados dos clientes, o processamento feito teve como objetivo facilitar a interpretação dos dados relativas às idades dos mesmos. Sendo assim, estes foram divididos em três categorias ("Young", "Middle Aged", "Elderly") como é possível verificar na figura abaixo:

```
> GroupAge <- cut(dataClients$Age, breaks= c(0, 30, 50, +Inf), labels= c("Young", "Middle Aged", "Elderly"),
 right=FALSE, ordered_result=TRUE)
> table(GroupAge)
GroupAge
      Young Middle Aged     Elderly
       1939        7869        7453
```

Figura 2 – Processamento das Idades dos Clientes

3.2 Dataset Purchases

Já no caso dos dados relativos às vendas, foram primeiro analisados os atributos numéricos:

```
> summary(numericAttrs) # ver o range de valores numericos
      YEAR          DOCNUMBER           LIN              QNT
 Min.   :2011   Min.   :    1   Min.   : 1.000   Min.   : 1.00
 1st Qu.:2012   1st Qu.: 6115   1st Qu.: 1.000   1st Qu.: 1.00
 Median :2013   Median :12798   Median : 2.000   Median : 1.00
 Mean   :2013   Mean   :18212   Mean   : 2.218   Mean   : 1.07
 3rd Qu.:2013   3rd Qu.:24167   3rd Qu.: 3.000   3rd Qu.: 1.00
 Max.   :2014   Max.   :79564   Max.   :29.000   Max.   :52.00
 TABLEUNITPRICE   DESCOUNTPERCENT1      PACKAGE    PACKAGEFACT      TOTAL
 Min.   :  0.36   Min.   :0.00e+00   Min.   :1   Min.   :1    Min.   :  0.40
 1st Qu.:  2.40   1st Qu.:0.00e+00   1st Qu.:1   1st Qu.:1    1st Qu.:  2.45
 Median :  3.72   Median :0.00e+00   Median :1   Median :1    Median :  3.87
 Mean   : 11.51   Mean   :4.71e-04   Mean   :1   Mean   :1    Mean   : 11.35
 3rd Qu.:  9.95   3rd Qu.:0.00e+00   3rd Qu.:1   3rd Qu.:1    3rd Qu.: 10.98
 Max.   :858.50   Max.   :1.50e+01   Max.   :1   Max.   :1    Max.   :707.00
```

Figura 3 – Sumário atributos numéricos das Vendas

```
> correlationMatrix
                        YEAR    DOCNUMBER          LIN          QNT TABLEUNITPRICE DESCOUNTPERCENT1 PACKAGE PACKAGEFACT        TOTAL
YEAR            1.0000000000 -0.192764369 0.0068406484  6.863371e-03   0.007001528   -5.302407e-03      NA          NA 0.0002316043
DOCNUMBER      -0.1927643690  1.000000000 0.0181792738 -6.843625e-03   0.063818872   -2.134971e-03      NA          NA 0.0627724083
LIN             0.0068406484  0.018179274 1.0000000000  1.412160e-02  -0.130983593    7.242585e-04      NA          NA -0.1393574587
QNT             0.0068633731 -0.006843625 0.0141215985  1.000000e+00  -0.067247813   -1.395881e-05      NA          NA 0.0052914348
TABLEUNITPRICE  0.0070015284  0.063818872 -0.1309835928 -6.724781e-02   1.000000000   -2.873736e-03      NA          NA 0.9843003719
DESCOUNTPERCENT1 -0.0053024070 -0.002334971 0.0007242585 -1.395881e-05  -0.002873736    1.000000e+00      NA          NA -0.0030616184
PACKAGE                 NA          NA          NA           NA            NA              NA          1          NA           NA
PACKAGEFACT             NA          NA          NA           NA            NA              NA         NA           1           NA
TOTAL           0.0002316043  0.062772408 -0.1393574587  5.291435e-03   0.984300372   -3.061618e-03      NA          NA 1.0000000000
```

Figura 4 – Matriz de correlação

Como é possível afirmar, os atributos PACKAGE e PACKAGEFACT são irrelevantes para os estudos uma vez que são constantes ao logo dos dados. Desta forma, ambos os atributos foram removidos do dataframe.

Por fim, ao analisar o atributo DEPARTMENT, verificou-se que este atributo possui classes muito mal distribuídas.

```
> table(dataPuchases$DEPARTMENT) # muito mais distrubidas as classes

   Acessorios      Cosmetica       Estetica NO DEPARTMENT     Perfumaria       Toucador 
         3256           50463               6              26           37359          216861 
```

Figura 5 – Distribuição do Atributo DEPARTAMENT

Como podemos afirmar, a Estética e NO DEPARTMENT possuem valores bastante inferiores relativo ao resto das classes. Assim, após analisar as vendas efetuadas com produtos destes departamentos, foi possível concluir que:

1. As vendas do departamento de estética são produtos de maquilhagem e manicure;
2. As vendas de produtos sem departamentos são maioritariamente cremes e máscaras.

Desta forma, ambos fazem sentido serem adicionados ao departamento de cosmética. Assim, o número de classes do atributo foi reduzido e o atributo encontra-se mais concentrado.

Outro atributo que se encontra mal distribuído entre as suas classes é o CODDOC. No entanto, como o grupo não consegui perceber a distinção entre os seus valores, não foram feitos os ajustes necessários a esta classe, contudo este é um ponto a ser melhorado para a segunda entrega do trabalho.

4. Exploração dos dados

Nesta secção é apresentada a exploração gráfica que foi feita aos dados, de forma a ser possível compreender de forma genérica os clientes da rede de perfumarias e os seus hábitos de compra. Cada subsecção terá a análise de um ou mais gráficos que foram considerados pertinente.

4.1 Exploração do perfil dos Clientes

Um dos grandes objetivos desta análise, é inferir quem é/são o(s) cliente(s) modelo do grupo de perfumarias. Para tal, explorou-se o género dos clientes (Figura 6).

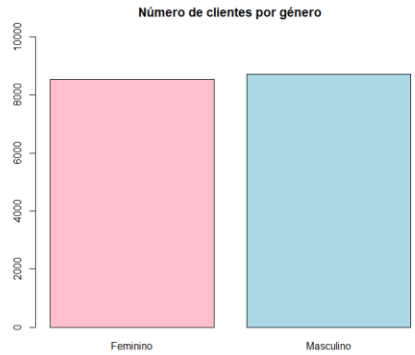


Figura 6 - Comparação do número de Clientes por Género

O objetivo na criação deste gráfico é verificar a quantidade de pessoas de cada género que existe no conjunto de dados. Este estudo é relevante porque, sendo o género um campo conhecidamente tendencioso, nomeadamente nos modelos de inteligência artificial[1], garante que , pela parecença dos números entre géneros, não existirá esta problemática. Por outro lado, conseguimos concluir que existem clientes dos dois géneros em grande quantidade. O código utilizado para obter este gráfico é:

```
barplot(table(dataClients$Gender),
        ylim=c(0,10000),
        main="Número de clientes por género",
        col=c("pink","lightblue"),
        names.arg=c("Feminino","Masculino")
)
```

Figura 7 – Código de gráfico de comparação do número de Clientes por Género

Para além do género, foi também explorada a idade dos clientes, representada na figura 8.

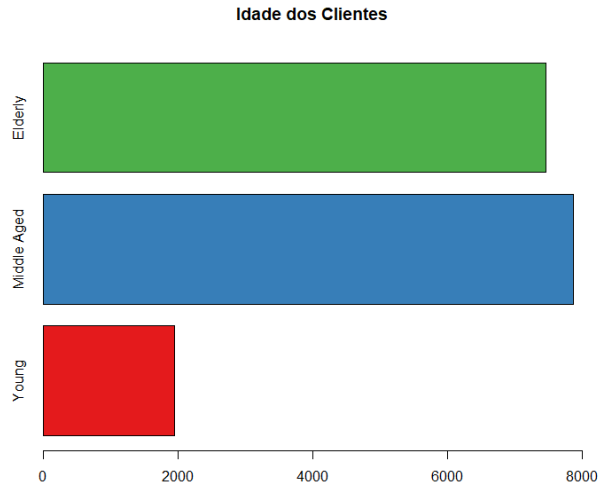


Figura 8 - Comparação do número de Clientes por Idade

Este estudo ajuda a conhecer melhor os clientes do grupo, percebendo-se que grande parte dos clientes são pessoas de meia

idade e idosos. Para reproduzir este gráfico, o código necessário ficará a baixo:

```
# Gráfico de comparação de Idades dos clientes
barplot(table(dataClients$GroupAge),
main="Idade dos Clientes",
col=coul,
horiz=T,
xlim =c(0,8000)
)
```

Figura 9 – Código de gráfico de comparação do número de Clientes por Idade

Para corroborar a interpretação feita baseada nestes dois gráficos, foi desenvolvido um gráfico de densidade da Idade por Género, representado na figura 10.

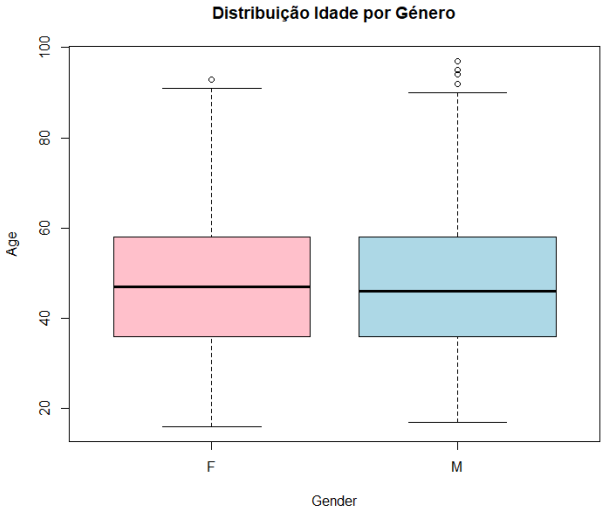


Figura 10 – Distribuição idade por género

Com este gráfico podemos perceber, mais uma vez, que a maior densidade de idades está entre os 40 e os 60 anos, como pudemos observer pelas duas faixas etárias mais presentes no gráfico anterior. Por outro lado, confirma-se também que, para cada género, o número de pessoas e a idade das mesmas que frequentam as lojas do grupo é a mesma. O código para reproduzir este gráfico é:

```
boxplot(dataClients$Age ~ dataClients$Gender,
 col=c("pink","lightblue")
 , main="Distribuição Idade por Genero",
 xlab="Gender",
 ylab="Age")
```

Figura 11 – Código de gráfico de distribuição da idade por género

Estudou-se também o número de filhos que cada cliente tem. O resultado desse estudo está apresentado na figura 12.

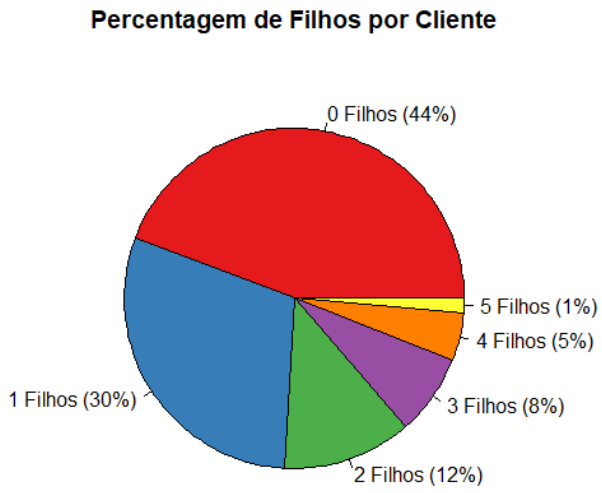


Figura 12 – Percentagem de filhos por cliente

Podemos perceber que maior parte dos clientes da empresa têm entre 0 e 2 filhos. O código para reprodução deste gráfico é apresentado na figura 13.

```
# Percentagem de Filhos por Cliente
par(mfrow=c(1,1))

slices <- table(dataClients$NumChildren)
lbls <- rownames(slices)
lbls <- paste(lbls, "Filhos (")
pct <- round(slices / sum(slices) * 100)
lbls <- paste(lbls, pct, sep = "")
lbls <- paste(lbls, "%)", sep = "")

pie(slices, labels = lbls, col = coul,
 main = "Percentagem de Filhos por Cliente")
```

Figura 13– Código de gráfico de percentagem de filhos por cliente

Conseguimos também explorar , por zona, qual idade e género dos clientes do grupo. Esta exploração está representada no conjunto de gráficos a baixo:

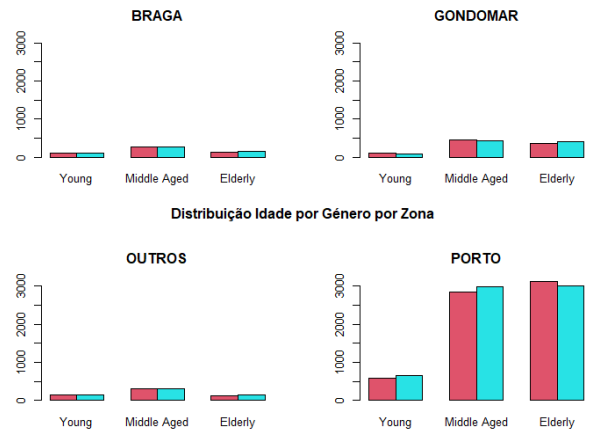


Figura 14- Gráfico de distribuição de idade por género e zona

É possível interpretar, em primeiro lugar, que maior parte dos clientes da loja são da cidade do Porto, sendo eles de que idade sejam. Existe, em Gondomar, uma massa de pessoas de meia idade significativa, apesar de bastante menos relevante do que os clientes do porto. O código para obter este gráfico está na figura a baixo.

```
par(mfrow=c(2,2)) # matriz 2 por 2 para apresentar graficos

for (zone in sort(unique(dataClients$Zone))) {
  rowsForZone <- dataClients[dataClients$Zone == zone, ]
  boxplot(rowsForZone$Age ~ rowsForZone$Gender,
          col=c(2,5), main=zone,
          xlab="Gender",
          ylab="Age")
}
mtext(expression(paste(bold("Distribuição Idade por Género por Zona"))),
      = 3,
      = -17.3,
      outer = TRUE)
```

Figura 15- Código de gráfico de distribuição de idade por género e zona

4.2 Exploração de hábitos de compra

Para além do perfil dos clientes, é importante perceber quais os hábitos de compra dos mesmos nas lojas do grupo. Esta subsecção apresenta o estudo estatístico nesta temática de hábitos de compra.

Em primeiro lugar foi considerado interessante perceber que produtos é que as pessoas de dada faixa etária compram. A figura abaixo apresenta o resultado deste interesse.

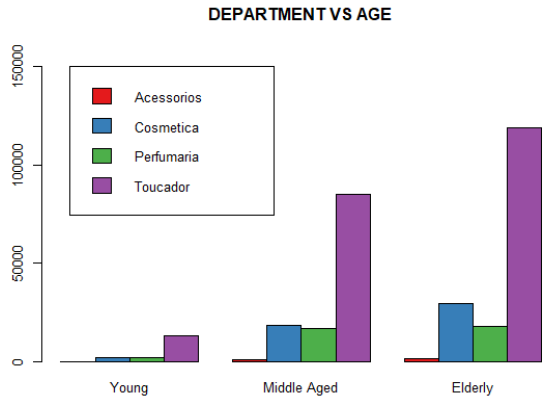


Figura 16 – Distribuição idade por género

Podemos observar que os produtos mais vendidos são da categoria "Toucador", independentemente da faixa etária. Podemos também observar que, enquanto as pessoas de meia-idade compram quase tanta perfumaria como cosméticos, as pessoas de faixa etária mais avançada compram substancialmente mais cosméticos do que perfumes. O código para reproduzir o gráfico acima está na figura abaixo.

```
barplot(table(dataMerged$DEPARTMENT,
dataMerged$GroupAge),
beside = T,
col=coul,
ylim = c(0,160000),
main= "DEPARTMENT VS AGE")

legend(list(x = 1.25,y = 150000),
levels(dataMerged$DEPARTMENT),
fill=coul)
```

Figura 17 – Código de gráfico de distribuição idade por género

Quanto a vendas do grupo, foi feito um estudo acerca do volume de vendas total e por ano, com divisão por trimestre. Para tal, primeiro construiu-se um gráfico de número de vendas por ano, sem considerar trimestres.

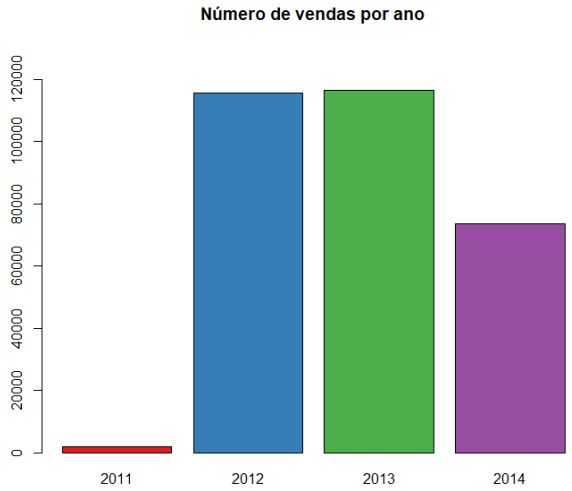


Figura 18 – Número de vendas por ano

Este gráfico mostra-nos que o volume de vendas registado no dataset para 2011 é pouco significativo, comparativamente aos outros anos. Isto deve-se apenas um trimester estar registado no dataset. Tal também acontece com 2014 onde só os três primeiros trimestres estão registados. Podemos observar este comportamento nos gráficos abaixo:

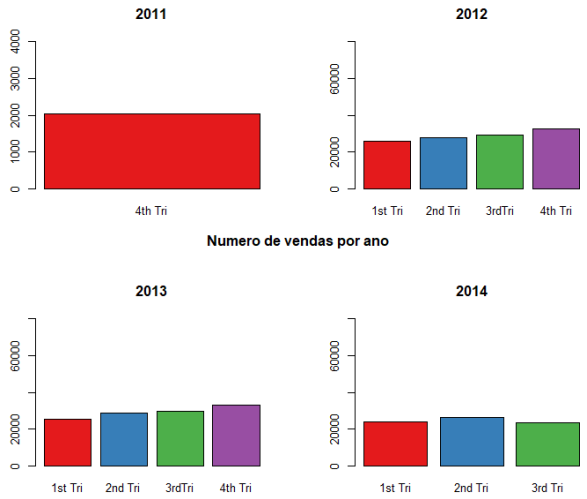


Figura 19 – Número de vendas por ano por trimestre

Podemos observar que os anos que estão completos (2012 e 2013) têm o seu expoente de vendas no último trimestre do ano, o que pode ser compreendido por esta altura corresponder a uma altura de muita compra (o Natal), os gráficos de 2011 e 2014, como não têm o ano completo, não é possível perceber a progressão para verificar se esta está igual. O código para reproduzir estes dois conjuntos de gráficos é:

```
purchasesByYear <- count(dataPuchases,"YEAR")
barplot(as.matrix(purchasesByYear)[,2],
main = "N□mero de vendas por ano",
col=coul,
ylim=c(0,150000),
names.arg=c("2011","2012","2013","2014"))
```

Figura 20 – Código para gráfico de número de vendas por ano

```
library(lubridate)
par(mfrow=c(2,2))
#2011
AKA <-ymd_hms(dataPuchases$DATE[dataPuchases$YEAR=="2011"])
ae <-quarter(AKA,
             with_year = FALSE)
barplot(table(ae),
        col=coul,
        ylim=c(0,4000),
        names.arg=c("4th Tri"),
        main="2011")

#2012
AKA <-ymd_hms(dataPuchases$DATE[dataPuchases$YEAR=="2012"])
ae <-quarter(AKA,
             with_year = FALSE)
barplot(table(ae),
        col=coul,
        ylim=c(0,80000),
        names.arg=c("1st Tri","2nd Tri","3rdTri","4th Tri"),
        main="2012")
#2013
AKA <-ymd_hms(dataPuchases$DATE[dataPuchases$YEAR=="2013"])
ae <-quarter(AKA,
             with_year = FALSE)
barplot(table(ae),
        col=coul,
        ylim=c(0,80000),
        names.arg=c("1st Tri","2nd Tri","3rdTri","4th Tri"),
        main="2013")
#2014
AKA <-ymd_hms(dataPuchases$DATE[dataPuchases$YEAR=="2014"])
ae <-quarter(AKA,
             with_year = FALSE)
barplot(table(ae),
        col=coul,
        ylim=c(0,80000),
        names.arg=c("1st Tri","2nd Tri","3rd Tri"),
        main="2014")
mtext(expression(paste(bold("Numero de vendas por ano"))),
      side = 3,
      line = -19,
      outer = TRUE)
```

Figura 21 – Código para gráfico de número de vendas por ano por trimestrex

Decidiu-se também perceber, por ano, quais os departamentos que mais vendiam. Para tal , produziu-se o gráfico a baixo apresentado:

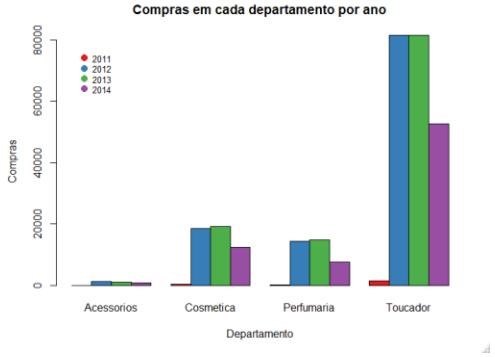


Figura 22 – Gráfico de compras em cada departamento por ano

Podemos, mais uma vez observer que, independentemente do ano, Toucador é o departamento que mais vende. O código para obter este gráfico fica na image a baixo:

```
# Compras em cada departamento do ano
coul2 <- brewer.pal(4, "Set1")
barplot(table(dataMerged$YEAR,dataMerged$DEPARTMENT),
        main = "Compras em cada departamento por ano",
        ylab = "Compras", xlab = "Departamento",
        col = coul2,
        beside = TRUE)
legend("topleft",
       legend = c("2011","2012","2013","2014"),
       col = coul2,
       bty = "n",
       pch=20 ,
       pt.cex = 2,
       cex = 0.8,
       horiz = FALSE,
       inset = c(0.05, 0.05))
```

Figura 23 – Código de gráfico de compras em cada departamento por ano

4.3 Análise de índice RFM

O índice RFM (Recency, Frequency & Monetary) é uma análise baseada em comportamentos usada para segmentar clientes, examinando o histórico de transações destes numa empresa, tais como o quão recentemente um cliente comprou na loja, o quão regularmente estes compram na loja e quanto compram na loja. Este índice é baseado na ideologia de marketing que 80% do lucro de uma loja vem de 20% dos clientes e ajuda a identificar os clientes que melhor irão responder a promoções, segmentando-os em várias categorias. Este índice foi considerado ideal para fazer clustering dos clientes, dividindo-os em várias categorias. Para tal, foi utilizado o package rfm, estando o código utilizado para tal na figura abaixo:

```
rfmDataAggregate <- aggregate(dataMerged["TOTAL"],
                              by=list(dataMerged$Client,
                              dataMerged$DATE), FUN=sum)
rfmData <- data.frame(customer_id = rfmDataAggregate$Group.1,
                      order_date = rfmDataAggregate$Group.2,
                      revenue = rfmDataAggregate$TOTAL)
rfmData$order_date <- as.Date(rfmData$order_date)
rfmData$revenue <- as.numeric(rfmData$revenue)
analysis_date <-  lubridate::as_date(max(rfmData$order_date), tz= "UTC")
```

Figura 24 – Código de cálculo de rfm

Após este cálculo, é necessário interpretar os valores devolvidos por este package. Para tal, foram realizados alguns estudos estatísticos sobre os dados, nomeadamente o efeito da Recency e Frequency no valor monetário gasto pelos clientes. Para tal foi efetuado um heatmap que mostra este efeito (Figura 25).

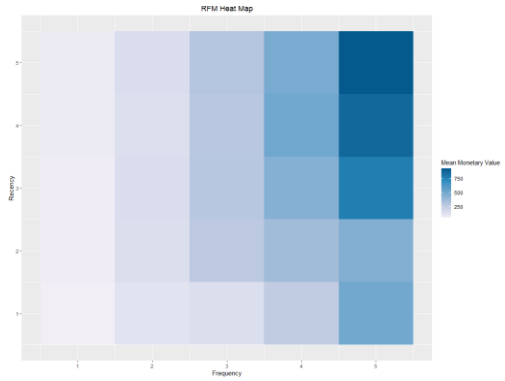


Figura 25 – Hashmap de efeito da recency e frequency no valor monetário dos clientes

Podemos observar que, como esperado, quem vai mais frequentemente e foi mais recentemente à loja, é quem gasta mais dinheiro. Para além disso, podemos confirmar que a Frequency afeta mais o valor monetário gasto do que a Recency.

Para além disso, foram feitas analises a cada par de variáveis. Os gráficos dessas analises estão apresentados nas figuras a baixo.

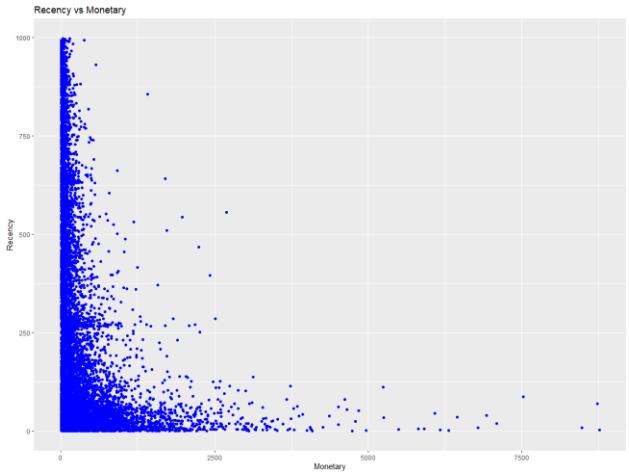


Figura 26 – Gráfico de comparação de Recency Vs Monetary

Por este gráfico, podemos observar que os clientes que visitaram a loja mais recentemente geraram mais lucro do que os que visitaram num passado mais distante.

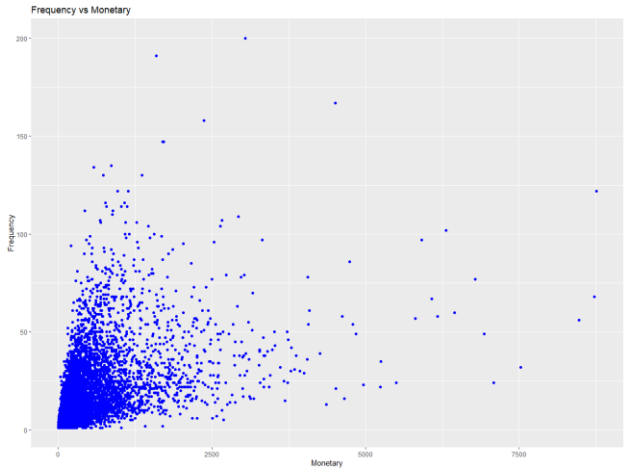


Figura 27 – Gráfico de comparação de Frequency Vs Monetary

Este gráfico mostra-nos que, ao aumentar a frequência, o valor monetário também aumenta.

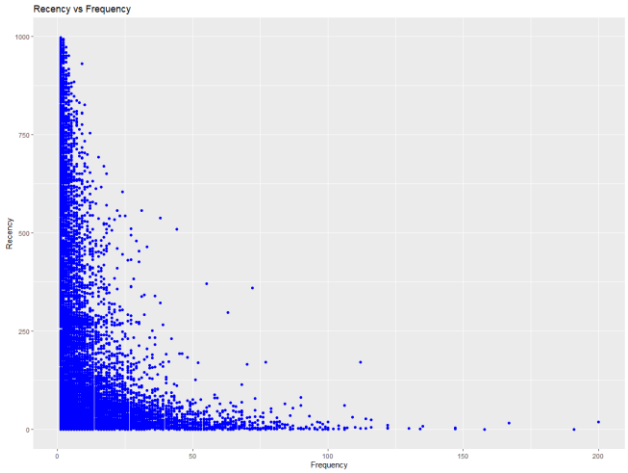


Figura 28 – Gráfico de comparação de Recency Vs Frequency

Por ultimo, este gráfico mostra-nos que as pessoas que visitaram num passado distante têm pouca frequência. Sendo assim, maior frequência está relacionada com visitas mais recentes.

5. Segmentação dos clientes

Dado o índice mencionado anteriormente, foi feita a segmentação dos clientes utilizando-o em conjunto com os dados dos mesmos. Para tal, foi utilizado o algoritmo de segmentação K-means. Este algoritmo, que depende do valor que atribuímos ao K, irá devolver em que conjunto está cada cliente. Para calcular qual será o melhor K para o algoritmo, utilizou-se o coeficiente de Silhouette. O K para qual o coeficiente de Silhouette for mais próximo de 1 será o mais adequado. Para tal foi utilizado o código apresentado em baixo.

```
dist <- dist(dataClustering.std)
sil <- silhouette(dataClustering.k3$cluster, dist)
summary(sil)
plot(sil)
```

Figura 25 – Código de cálculo de coeficiente de silhouette

O K que foi encontrado como o melhor foi K=3. Sendo assim, foi utilizado o K-means com o código da figura a baixo para fazer o clustering dos dados.

```
dataClustering.k3 <- kmeans(dataClustering.std, centers=3, iter.max=100, nstart = 25)
```

Figura 26 – Código de cálculo de K-means para K=3

O resultado deste algorítmo está descriminado no gráfico a baixo, mostrando o volume de cada um dos 3 clusters:

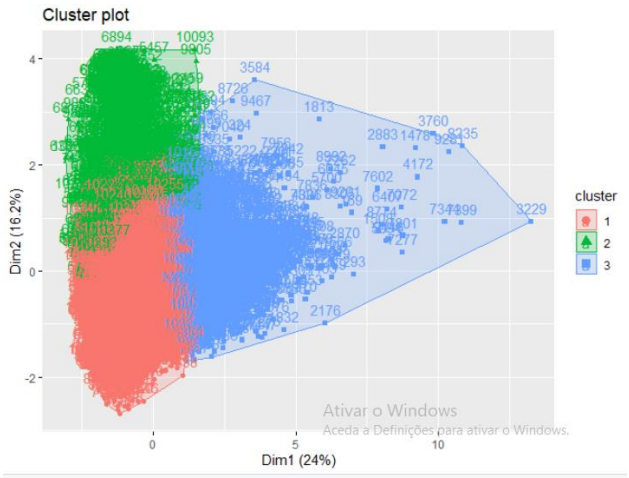


Figura 27 – Gráfico de resultado de clustering dos dados

Podemos observar que o Cluster número 3 (azul) tem um volume maior do que os outros dois, motivo que se verifica devido a haver valores mais dispersos.

Iremos agora avaliar quais as diferenças entre os clientes correspondentes a cada segmento. Para tal, serão apresentados conjuntos de gráficos, relativos á mesma informação mas para cada um dos clusters, de forma a comparer as propriedades de cada um. Em primeiro lugar, fez-se uma análise quanto ao género dos clientes de cada segmento. Os gráficos relativos a essa análise são apresentados na figura 28.

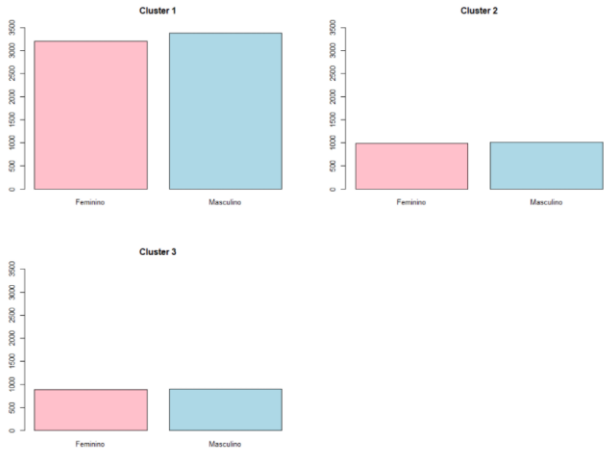


Figura 27 – Género dos clientes de cada cluster

Através destes gráficos, é possível tirar diferentes conclusões. A primeira relaciona-se com a separação por clusters, que indica uma maior presença de clientes no cluster 1. De seguida, é possível notar que nos três gráficos a quantidade absoluta de pessoas por género é muito semelhante pelo que se conclui, principalmente considerando o cluster mais populado (1), que o género não é uma da(s) variável(eis) determinantes para a divisão por clusters.

De seguida, foi também analisada a idade dos clientes de cada segmento. Os gráficos correspondentes a esta análise são apresentados na figura abaixo.

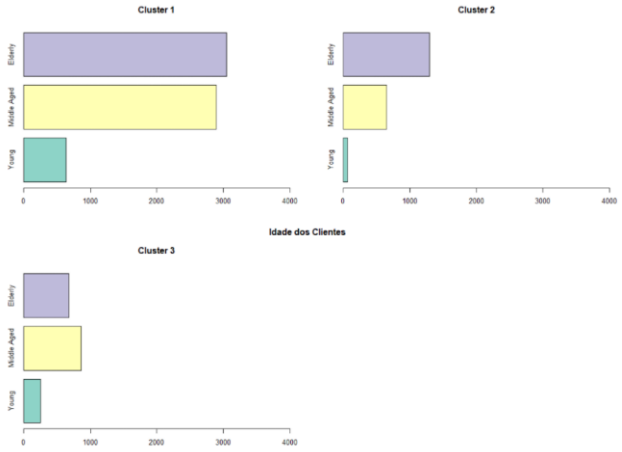


Figura 28 – Idade dos clientes de cada cluster

Este conjunto de gráficos divide cada cluster por idades (Young, Middle Aged e Elderly). Mais uma vez, podemos concluir que o cluster 1 é aquele em que mais clientes se enquadram. Pela diversidade de clientes, em relação à idade, presentes em cada um destes cluster, é notória que a idade não é uma variável relevante para a divisão de clientes por clusters.

Para além destas análises foi também efetuada uma análise á zona de cada cliente, segmentado pelo género. O gráfico para este estudo está apresentado nas imagens a baixo.

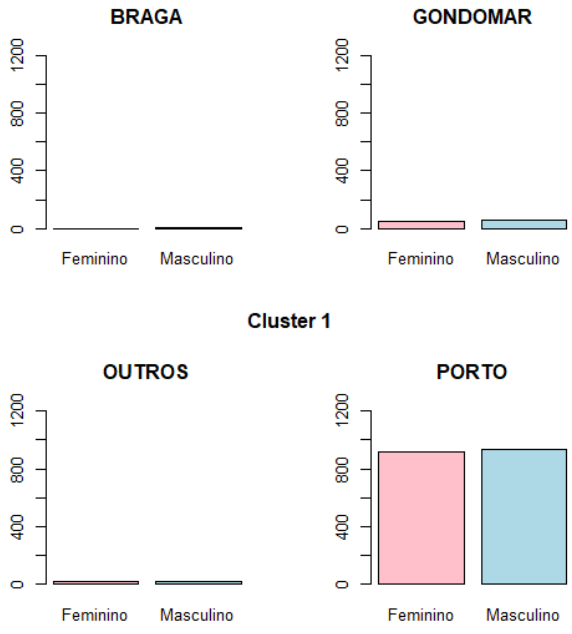


Figura 29 – Zonas por idade e género de cada cliente do cluster 1

~

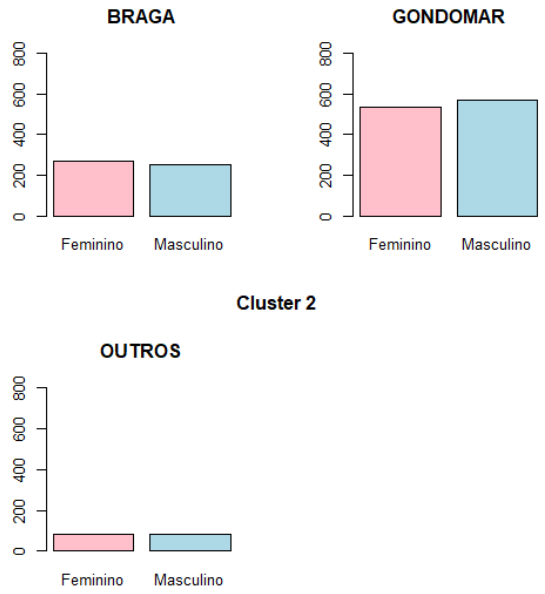


Figura 30 – Zonas por idade e género de cada cliente do cluster 2

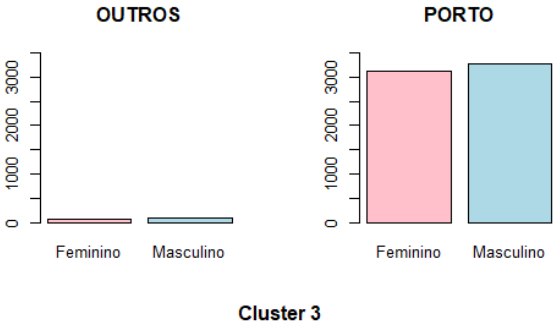


Figura 31 – Zonas por idade e género de cada cliente do cluster 3

Por estes gráficos, podemos observar que a zona é um fator de decisão no que diz respeito a classificar como do segmento 3, visto que estes apenas têm clientes do setor “Porto” e “Outros”.

Por ultimo, foi feito um estudo relativamente ás diferenças entre número de filhos para os clientes de cada segmento. Os gráficos para verificação são apresentados nas figuras a baixo.

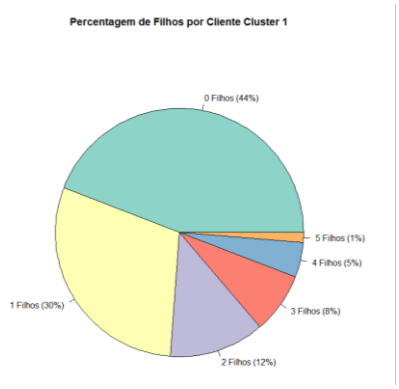


Figura 32 – Percentagem de filhos por cliente para cluster 1

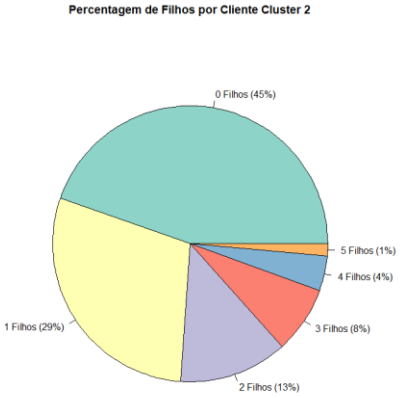


Figura 33 – Percentagem de filhos por cliente para cluster 2

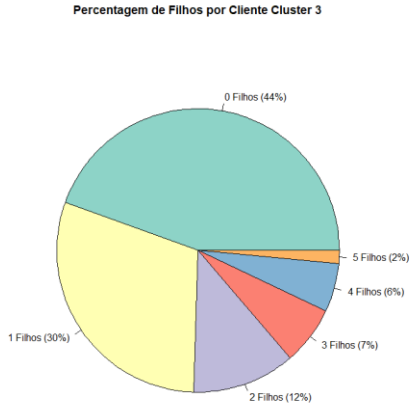


Figura 34 – Percentagem de filhos por cliente para cluster 3

Podemos daqui concluir que, visto que as diferenças percentuais são baixas entre os segmentos, a idade não é um fator muito significativo na diferenciação entre os vários segmentos.

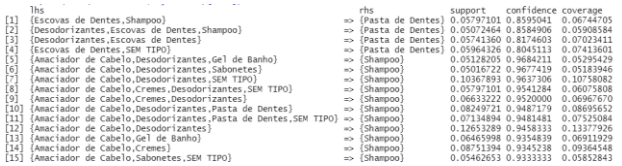
6. Regras de associação

Esta secção apresenta regras de associação utilizando o algoritmo apriori para cada um dos segmentos apresentados anteriormente. O código para utilizar este algoritmo em R está apresentado na imagem abaixo.

```
# Com os dados anteriormente carregados crie um objecto basket
basket <- as(split(as.vector(dataMergedForCluster1$PRODUCTYPE), as.vector(dataMergedForCluster1$Client)), "transactions")
inspect(basket[1:5])

# Visualize a frequência dos itens numericamente
itemFreq <- itemFrequency(basket)
itemFreq
itemFrequency(basket[,1:3])
# Contabilize o nº de vezes em que cada item aparece
itemCount <- (itemFreq / sum(itemFreq))*sum(size(basket))
itemCount

# Aplique o algoritmo Apriori para extração de Regras de Associação com Supmin=5% e Confmin = 80%
basketRules <- apriori(basket, parameter = list(support=0.05, confidence=0.80, minlen=2))
summary(basketRules)

# Para o conjunto de regras anteriormente gerado visualize:

# 1. a gama de valores das medidas Coverage, Conviction e Leverage
measures <- interestMeasure(basketRules,
                            measure = c("coverage", "leverage", "conviction"),
                            transactions = basket)
measures
```

Figura 28 – Código de aplicação do algorimo apriori para um cluster

Algumas das regras encontradas, para cada cluster, estão apresentadas nas figuras a baixo.



	lhs		rhs	support	confidence	coverage
[1]	{Escovas de Dentes,Shampoo}	=>	{Pasta de Dentes}	0.05797101	0.8595041	0.06744705
[2]	{Desodorizantes,Escovas de Dentes,Shampoo}	=>	{Pasta de Dentes}	0.05072464	0.8584906	0.05908584
[3]	{Desodorizantes,Escovas de Dentes}	=>	{Pasta de Dentes}	0.05741360	0.8174603	0.07023411
[4]	{Escovas de Dentes,SEM TIPO}	=>	{Pasta de Dentes}	0.05964326	0.8045113	0.07413601
[5]	{Amaciador de Cabelo,Desodorizantes,Gel de Banho}	=>	{Shampoo}	0.05128205	0.9684211	0.05295429
[6]	{Amaciador de Cabelo,Desodorizantes,Sabonetes}	=>	{Shampoo}	0.05016722	0.9677419	0.05183946
[7]	{Amaciador de Cabelo,Desodorizantes,SEM TIPO}	=>	{Shampoo}	0.10367893	0.9637306	0.10758082
[8]	{Amaciador de Cabelo,Cremes,Desodorizantes,SEM TIPO}	=>	{Shampoo}	0.05797101	0.9541284	0.06075808
[9]	{Amaciador de Cabelo,Cremes,Desodorizantes}	=>	{Shampoo}	0.06633222	0.9520000	0.06967670
[10]	{Amaciador de Cabelo,Desodorizantes,Pasta de Dentes}	=>	{Shampoo}	0.08249721	0.9487179	0.08695652
[11]	{Amaciador de Cabelo,Desodorizantes,Pasta de Dentes,SEM TIPO}	=>	{Shampoo}	0.07134894	0.9481481	0.07525084
[12]	{Amaciador de Cabelo,Desodorizantes}	=>	{Shampoo}	0.12653289	0.9458333	0.13377926
[13]	{Amaciador de Cabelo,Gel de Banho}	=>	{Shampoo}	0.06465998	0.9354839	0.06911929
[14]	{Amaciador de Cabelo,Cremes}	=>	{Shampoo}	0.08751394	0.9345238	0.09364548
[15]	{Amaciador de Cabelo,Sabonetes,SEM TIPO}	=>	{Shampoo}	0.05462653	0.9333333	0.05852843

Figura 29 – 15 Regras fortes para Cluster 1

	lhs		rhs	support	confidence	coverage
[1]	{Escovas de Dentes,Shampoo}	=>	{Pasta de Dentes}	0.05797101	0.8595041	0.06744705
[2]	{Desodorizantes,Escovas de Dentes,Shampoo}	=>	{Pasta de Dentes}	0.05072464	0.8584906	0.05908584
[3]	{Desodorizantes,Escovas de Dentes}	=>	{Pasta de Dentes}	0.05741360	0.8174603	0.07023411
[4]	{Escovas de Dentes,SEM TIPO}	=>	{Pasta de Dentes}	0.05964326	0.8045113	0.07413601
[5]	{Amaciador de Cabelo,Desodorizantes,Gel de Banho}	=>	{Shampoo}	0.05128205	0.9684211	0.05295429
[6]	{Amaciador de Cabelo,Desodorizantes,Sabonetes}	=>	{Shampoo}	0.05016722	0.9677419	0.05183946
[7]	{Amaciador de Cabelo,Desodorizantes,SEM TIPO}	=>	{Shampoo}	0.10367893	0.9637306	0.10758082
[8]	{Amaciador de Cabelo,Cremes,Desodorizantes,SEM TIPO}	=>	{Shampoo}	0.05797101	0.9541284	0.06075808
[9]	{Amaciador de Cabelo,Cremes,Desodorizantes}	=>	{Shampoo}	0.06633222	0.9520000	0.06967670
[10]	{Amaciador de Cabelo,Desodorizantes,Pasta de Dentes}	=>	{Shampoo}	0.08249721	0.9487179	0.08695652
[11]	{Amaciador de Cabelo,Desodorizantes,Pasta de Dentes,SEM TIPO}	=>	{Shampoo}	0.07134894	0.9481481	0.07525084
[12]	{Amaciador de Cabelo,Desodorizantes}	=>	{Shampoo}	0.12653289	0.9458333	0.13377926
[13]	{Amaciador de Cabelo,Gel de Banho}	=>	{Shampoo}	0.06465998	0.9354839	0.06911929
[14]	{Amaciador de Cabelo,Cremes}	=>	{Shampoo}	0.08751394	0.9345238	0.09364548
[15]	{Amaciador de Cabelo,Sabonetes,SEM TIPO}	=>	{Shampoo}	0.05462653	0.9333333	0.05852843

Figura 30 – 15 Regras fortes para Cluster 2

	lhs		rhs	support	confidence	coverage
[1]	{Escovas de Dentes,Shampoo}	=>	{Pasta de Dentes}	0.05797101	0.8595041	0.06744705
[2]	{Desodorizantes,Escovas de Dentes,Shampoo}	=>	{Pasta de Dentes}	0.05072464	0.8584906	0.05908584
[3]	{Desodorizantes,Escovas de Dentes}	=>	{Pasta de Dentes}	0.05741360	0.8174603	0.07023411
[4]	{Escovas de Dentes,SEM TIPO}	=>	{Pasta de Dentes}	0.05964326	0.8045113	0.07413601
[5]	{Amaciador de Cabelo,Desodorizantes,Gel de Banho}	=>	{Shampoo}	0.05128205	0.9684211	0.05295429
[6]	{Amaciador de Cabelo,Desodorizantes,Sabonetes}	=>	{Shampoo}	0.05016722	0.9677419	0.05183946
[7]	{Amaciador de Cabelo,Desodorizantes,SEM TIPO}	=>	{Shampoo}	0.10367893	0.9637306	0.10758082
[8]	{Amaciador de Cabelo,Cremes,Desodorizantes,SEM TIPO}	=>	{Shampoo}	0.05797101	0.9541284	0.06075808
[9]	{Amaciador de Cabelo,Cremes,Desodorizantes}	=>	{Shampoo}	0.06633222	0.9520000	0.06967670
[10]	{Amaciador de Cabelo,Desodorizantes,Pasta de Dentes}	=>	{Shampoo}	0.08249721	0.9487179	0.08695652
[11]	{Amaciador de Cabelo,Desodorizantes,Pasta de Dentes,SEM TIPO}	=>	{Shampoo}	0.07134894	0.9481481	0.07525084
[12]	{Amaciador de Cabelo,Desodorizantes}	=>	{Shampoo}	0.12653289	0.9458333	0.13377926
[13]	{Amaciador de Cabelo,Gel de Banho}	=>	{Shampoo}	0.06465998	0.9354839	0.06911929
[14]	{Amaciador de Cabelo,Cremes}	=>	{Shampoo}	0.08751394	0.9345238	0.09364548
[15]	{Amaciador de Cabelo,Sabonetes,SEM TIPO}	=>	{Shampoo}	0.05462653	0.9333333	0.05852843

Figura 30 – 15 Regras fortes para Cluster 3

Estas são algumas das regras que podem levar a promoções especializadas ou a re-organização da perfumaria de forma a que produtos que estão na mesma regra fiquem perto, para que seja facil para os clientes que compram um também levarem o outro.

Com estes dados, é possível observar quais os tipos de produtos mais frequentes nas regras. Gráficos representativos dessa informação são apresentados abaixo.

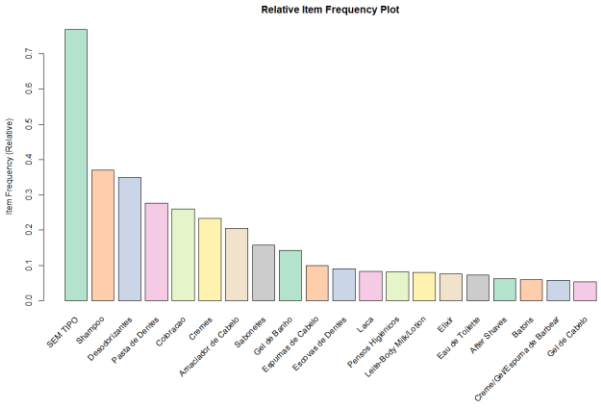


Figura 31 – Frequência Relativa dos Tipos de produtos nas regras para o Cluster 1

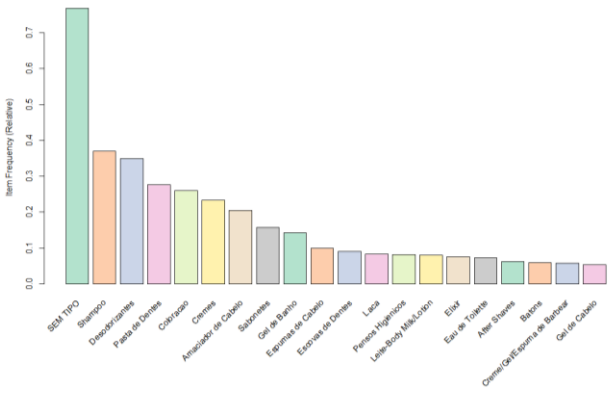


Figura 32 – Frequência Relativa dos Tipos de produtos nas regras para o Cluster 2

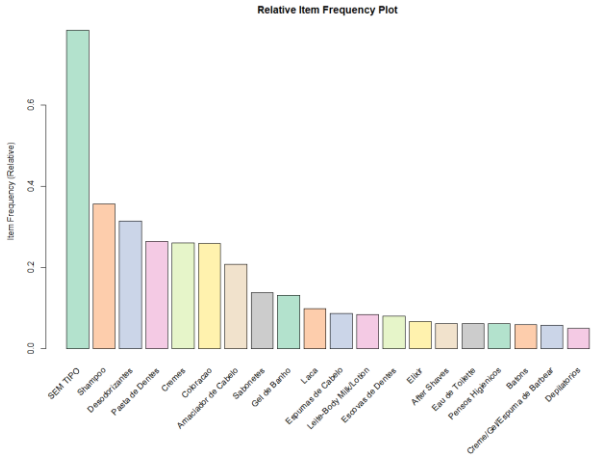


Figura 33 – Frequência Relativa dos Tipos de produtos nas regras para o Cluster 3

Podemos observar que, dependendo do segmento que observamos, a frequência dos items é diferente. Apesar disso, "SEM TIPO" é a mais regular de aparecer nas regras, possivelmente por serem produtos bastante variados no mesmo tipo de produto.

7. Previsão de segmento de novo cliente

Esta secção contêm a investigação relativamente a como podemos prever, dado um novo cliente que chega á loja, em que segmento este se pode vir a inserir, de forma a focar (ou não) as campanhas publicitárias a esse utilizador e qual seria a mais adequada para tal.

Para tal, foram explorados vários modelos de previsão de forma a classificar num dos 3 clusters identificados os novos clientes. Foram utilizados 70% dos dados de forma aleatória para treinar os modelos e 30% dos dados para teste do mesmo.

Cada subsecção desta secção terá a informação sobre um modelo e os seus resultados.

7.1 K vizinhos mais próximos

A classificação baseada em "vizinhos" é um tipo de "lazy learning" uma vez que não tenta construir um modelo interno geral, mas sim classifica utilizando apenas instâncias dos dados de treino armazenados [2].

O knn é um modelo que considera a proximidade entre dados na realização de previsões, assumindo que dados similares tendem a estar concentrados na mesma região no espaço de dispersão de dados.

A figura abaixo mostra como este algoritmo foi utilizado em R.

```
k <- c()
falsosPositivos <- c()
falsosNegativos <- c()
accuracy <- c()

  predictions <- knn(train[-5] , test[-5] , cl = train$Cluster, k = 3)

results <- rbind(results, allcrosstablemeasures(test$Cluster, predictions, modelname = "KNN"))
results
```

Figura 34 – Código da aplicação do modelo Knn

Os resultados do teste são apresentados na matriz de confusão a baixo:

Actual	Predicted 1	2	3	Row Total
1	33 0.011	34 0.011	535 0.172	602
2	9 0.003	501 0.161	28 0.009	538
3	64 0.021	21 0.007	1889 0.607	1974
Column Total	106	556	2452	3114

Figura 35 – Matriz de confusão para o Knn

A partir desta matriz podemos calcular as métricas de avaliação que serão utilizadas para este e todos os modelos: accuracy, precision e recall. A accuracy obtida é de 77.5%. A precisão para cada classe é, de 1 a 3, 0.0548, 0.931 e 0.956. Já o Recall, também respetivamente de 1 a 3, foi de 0.311, 0.901, 0.770. Podemos então concluir que este modelo, apesar de ter uma taxa de acerto elevada, é pouco eficaz a classificar clientes como do cluster 1.

7.2 Naive Bayes

Este modelo de classificação de Naive Bayes baseia-se na suposição de independência entre cada par de recursos. De uma maneira geral, o classificador assume que a presença de um determinado valor numa classe não é relacionada com a presença dos valores das classes da mesma instância.

A figura abaixo mostra como este algoritmo foi utilizado em R.

```
bayesModel <- naiveBayes(Cluster~., data = train, laplace = 1)

NBpredict <- predict(bayesModel, newdata = test[-5], type="class")
```

Figura 36 – Código de aplicação do modelo Naive Bayes

Os resultados do teste são apresentados na matriz de confusão a baixo:

Actual	Predicted 1	2	3	Row Total
1	11 0.004	36 0.012	555 0.178	602
2	30 0.010	508 0.163	0 0.000	538
3	27 0.009	14 0.004	1933 0.621	1974
Column Total	68	558	2488	3114

Figura 37 – Matriz de confusão para o Naive Bayes

A accuracy obtida é de 78.7%. A precisão para cada classe é, de 1 a 3, 0.0182, 0.944 e 0.979. Já o Recall, também respetivamente de 1 a 3, foi de 0.161, 0.910 e 0.776. Podemos ver que, tal como o modelo anterior este é ineficaz a prever clientes para o cluster 1, classificando-os muito regularmente como clientes do cluster 3. Apesar disso, o modelo acaba por ter uma taxa de acerto bastante alta, visto que acerta com bastante pouca taxa de erro na classificação de elementos do cluster 3 e do cluster 2.

7.3 Support Vector Machine

O SVM é uma representação dos dados de treino como pontos no espaço separados por categorias. Depois é encontrada uma reta que separa os dados em n números de grupos de dados (dependendo do n número de classes) onde a distância ao ponto mais próximo de cada grupo é a maior possível. Essa reta vai servir como o nosso classificador para o grupo de dados de teste.

Foram utilizados 3 kernels diferentes na aplicação deste modelo, tendo assim 3 resultados diferentes.

7.3.1 Kernel Tanhdot

O kernel tanhdot utiliza a função de tangente hiperbólica para a classificação.

A figura abaixo mostra como este algoritmo foi utilizado em R.

```
SVNmodel <- ksvm(Cluster ~., data = train, kernel = "tanhdot", C=1, prob.model = TRUE)

SVNpredict <- predict(SVNmodel, newdata = test[-5])
```

Figura 38 – Código de aplicação do modelo SVM com kernel Tanhdot

Os resultados do teste para este modelo são apresentados na matriz de confusão a baixo:

	Predicted			
Actual	1	2	3	Row Total
1	157	56	389	602
	0.050	0.018	0.125	
2	28	365	145	538
	0.009	0.117	0.047	
3	450	113	1411	1974
	0.145	0.036	0.453	
Column Total	635	534	1945	3114

Figura 39 – Matriz de confusão para o modelo SVM com kernel Tanhdot

A accuracy obtida é de 62.1%. A precisão para cada classe é, de 1 a 3, 0.260, 0.678 e 0.715. Já o Recall, também respetivamente de 1 a 3, foi de 0.247, 0.684 e 0.725. Este modelo, comparativamente aos anteriores, tem uma taxa de acerto substancialmente mais baixa. É também menos preciso do que os anteriores.

7.3.2 Kernel Vanilladot

O kernel Vanilladot utiliza uma função linear para a classificação.

A figura abaixo mostra como este algoritmo foi utilizado em R.

```
SVNmodel <- ksvm(Cluster ~., data = train, kernel = "vanilladot", C=1, prob.model = TRUE)
SVNmodel

SVNpredict <- predict(SVNmodel, newdata = test[-5])
```

Figura 40 – Código de aplicação do modelo SVM com kernel Vanilladot

Os resultados do teste para este modelo são apresentados na matriz de confusão a baixo:

	Predicted			
Actual	1	2	3	Row Total
1	2	33	567	602
	0.001	0.011	0.182	
2	6	482	50	538
	0.002	0.155	0.016	
3	11	0	1963	1974
	0.004	0.000	0.630	
Column Total	19	515	2580	3114

Figura 41 – – Matriz de confusão para o modelo SVM com kernel Vanilladot

A accuracy obtida é de 78.6%. A precisão para cada classe é, de 1 a 3, 0.0033, 0.896 e 0.994. Já o Recall, também respetivamente de 1 a 3, foi de 0.105, 0.935 e 0.760. Este modelo tem uma taxa de acerto maior do que com o kernel apresentado anteriormente. Apesar disso tem (como todos os modelos apresentados anteriormente) muito pouco sucesso a classificar clientes do cluster 1.

7.3.3 Kernel Rbfdot

O kernel Rbfdot utiliza a função base radial gaussiana para a classificação. A forma de aplicar este algoritmo em R é a mesma do que os dois kernels apresentados anteriormente, apenas mudando o parametro kernel para "rbfdot".

Os resultados do teste para este modelo são apresentados na matriz de confusão a baixo:

	Predicted		
Actual	2	3	Row Total
1	40	562	602
	0.013	0.180	
2	511	27	538
	0.164	0.009	
3	17	1957	1974
	0.005	0.628	
Column Total	568	2546	3114

Figura 42 – – Matriz de confusão para o modelo SVM com kernel Rbfdot

Este modelo não classifica nenhum cliente como sendo da classe 1, motivo pelo qual esta não aparece nos valores previstos. Por este motivo, não se fez qualquer exploração sobre este modelo, visto que, claramente não é o modelo adequado para o problema. Apesar disso, foi decidido reportar a exploração deste kernel.

7.4 Árvores de Decisão e Random Forest

As árvores de decisão possuem recursos versáteis que ajudam a atualizar variáveis dependentes categóricas e contínuas. Este algoritmo divide o conjunto de dados da população em dois ou mais conjuntos homogéneos com base nos atributos mais significativos, tornando os atributos mais distintos e especializados possível.

O modelo Random Forest ajusta uma série de árvores de decisão em várias subamostras de conjuntos de dados e usa a média para

melhorar a previsão do modelo. O tamanho da subamostra é sempre igual ao tamanho da amostra de entrada, no entanto as amostras são retiradas com substituição[3].

7.4.1 Árvores de decisão

Foi utilizado o package C50 para treinar o modelo de árvore de decisão. O código para aplicar em R este modelo é apresentado na figura abaixo.

```
C5Model <- C5.0(train[-5], train$Cluster)
summary(C5Model)

C5pred <- predict(C5Model, test[-5])
```

Figura 43 – Código de aplicação do modelo de árvore de decisão

A aplicação deste modelo gerou a árvore seguinte:

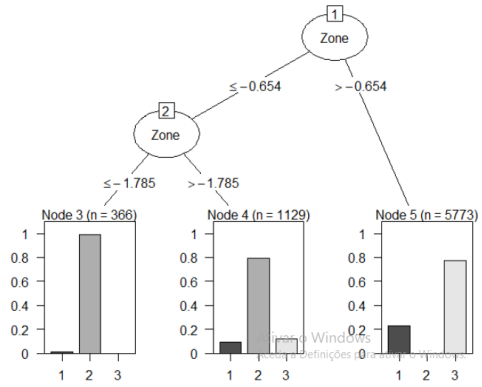


Figura 44 – Árvore gerada pelo modelo de árvore de decisão

Podemos concluir que o modelo considera que o único atributo importante para diferenciar os clientes é a zona onde estes moram (valores estão normalizados, motivo pelo qual correspondem a números). A matriz de confusão para este modelo está apresentada na figura 45.

Actual	Predicted 2	3	Row Total
1	33 0.011	569 0.183	602
2	482 0.155	56 0.018	538
3	0 0.000	1974 0.634	1974
Column Total	515	2599	3114

Figura 45 – Matriz de confusão para árvores de decisão

Podemos observar que, tal como o último modelo apresentado, este também não consegue classificar clientes como correspondendo ao cluster 1. Assim, este modelo também não foi explorado exaustivamente.

7.4.2 Random Forests

Foi utilizado o package randomForest para treinar o modelo. O código para aplicar este modelo é apresentado na figura abaixo. Foram utilizados os hiperparâmetros por defeito para o treino do modelo.

```
rf <- randomForest(
Cluster ~ ., data=train
)
pred = predict(rf, newdata=test[-5])
```

Figura 46 – Código de aplicação do modelo de random forests

A matriz de confusão para este modelo está apresentada na figura abaixo.

Actual	Predicted 1	2	3	Row Total
1	3 0.001	37 0.012	562 0.180	602
2	2 0.001	508 0.163	28 0.009	538
3	12 0.004	14 0.004	1948 0.626	1974
Column Total	17	559	2538	3114

Figura 47 – Matriz de confusão para modelo de Random Forests

A accuracy obtida é de 79.0%. A precisão para cada classe é, de 1 a 3, 0.005 ,0.944 e 0.987. Já o Recall, também respetivamente de 1 a 3, foi de 0.176, 0.909 e 0.768. Podemos então concluir que este modelo, apesar de ter uma taxa de acerto elevada, é, tal como os anteriormente referidos, pouco eficaz a classificar clientes como do cluster 1.

7.5 Rede Neuronal Artificial

As redes neuronais artificiais são uma parte da área de machine learning que se inspira na forma como funciona o cérebro humano para obter conhecimento. É composta por um conjunto de camadas, nomeadamente a camada de entrada, a camada de saída e as camadas escondidas. A camada de entrada recebe os valores de entrada da rede e com os quais vai obter conhecimento. As camadas escondidas computam uma função que retornará um resultado de uma previsão na camada de saída. Para tal, cada comunicação entre neurónios terá um peso definido, peso esse que será atualizado a cada iteração do treino até chegar ao resultado ótimo. O código para aplicar esse algoritmo com o package "nnet" em R está apresentado na figura abaixo.

```
fishing1<-nnet(Cluster~.,data=train,size=15, maxit=1500)

k<- predict(fishing1, test[-5],type = "class")
```

Figura 48 – Código para execução de modelo de rede neuronal artificial

Dado este modelo, foi gerada matriz de confusão apresentada na figura abaixo:

	Predicted			
Actual	1	2	3	Row Total
1	4	561	36	601
	0.001	0.180	0.012	
2	10	1938	26	1974
	0.003	0.623	0.008	
3	0	26	512	538
	0.000	0.008	0.164	
Column Total	14	2525	574	3113

Figura 49 – Matriz de confusão para modelo de rede neuronal artificial

A accuracy obtida é de 78.8%. A precisão para cada classe é, de 1 a 3, 0.0067, 0.982, 0.952. Já o Recall, também respetivamente de 1 a 3, foi de 0.286, 0.768 e 0.892. Mais uma vez podemos observar que, apesar da accuracy alta, o a precisão e o recall mostram que este modelo é pouco eficaz a classificar clientes do cluster 1.

8. Conclusões

Podemos observar que todos os modelos sucederam mal na classificação de clientes do cluster 1, apesar de terem todos uma exatidão alta. Desta análise podemos retirar que a exatidão (Accuracy) não é uma boa medida para avaliar modelos de classificação, dado que é possível ter níveis muito altos sem ter sucesso na classificação. Por outro lado, podemos observar que os dados existentes no dataset sobre os clientes (sem histórico de compras) não são suficientes para classificar com precisão o tipo (cluster) de cliente que estes poderão vir a ser. Possivelmente, havendo mais dados, os modelos poderiam ter tido um maior êxito.

Apesar disso, esta análise tem valor na separação de clientes já existentes pelo seu tipo, de forma que seja possível focar campanhas publicitárias mais personalizadas, dependendo do tipo de cliente.

9. Referências

[1] S. Leavy, G. Meaney, K. Wade, and D. Greene, "Mitigating gender bias in machine learning data sets," *Commun. Comput. Inf. Sci.*, vol. 1245 CCIS, no. May, pp. 12–26, 2020, doi: 10.1007/978-3-030-52485-2_2.

[2] "7 Types of Classification Algorithms - Analytics India Magazine." https://analyticsindiamag.com/7-types-classification-algorithms/ (accessed May 02, 2021).

[3] "Which one to use - RandomForest vs SVM vs KNN? - techniques - Data Science, Analytics and Big Data discussions." https://discuss.analyticsvidhya.com/t/which-one-to-use-randomforest-vs-svm-vs-knn/2897/3 (accessed May 02, 2021).