Movies Recommendation in Flixste

Inês Rocha, Alona Spasenko

16/05/2020

Introduction

Neste trabalho foram utilizados os modelos de recommendação: * Popularity * Association Rules * Collaborative Filtering * Context-aware recommendation Estes modelos foram aplicados a matrizes reais (matrizes com os ratings dos utilizadores) e a matrizes binárias (matrizes com indicação se um utilizador viu um certo filme).

Problem definition

Com a evolução do nosso mundo atual, é cada vez mais importante reter a atenção dos utilizadores, e uma das estratégias para obter isto é recomendar conteúdo que o utilizador goste. Dado a plataforme de filmes Flixer, vamos obter um sistema de recomendação que permita que os utilizadores recebam os melhores filmes que se enquadram com as suas preferências.

Exploratory data analysis and pre-processing steps

Pre-Processing steps

Ao ler os ficheiros txt, verificamos que o ficheiro "movie.txt" tinha uma vírgula separar o nome do filme do seu ID. Na primeira tentativa foi usada essa vírgula como o separador, mas apercebemo-nos que também existiam vírgulas nos títulos. Então para conseguirmos passar os dados para um tabela criámos uma expressão regular (",([^,]*)$","^\1") que encontrava a última vírgula (a vírgula que separa o nome do id) e substituía essa vírgula pelo caracter '^'.
Com esta substituição já podíamos na função read.table usar '^' como o separador e obter os dados corretos.

```
temp <-read.table(text = gsub(",([^,]*)$", "^\\1", readLines("movie.txt")),
                  header = TRUE, sep="^", fill = TRUE,comment.char = "",
                  na.strings = "?")
movies <- tbl_df(temp)
```

Depois de termos conseguido lido os ficheiros necessários foi feita uma limpeza aos dados:

1. Remover users que não tenham um destes géneros: "Female", "Male", "N/A"
2. Remover a data e a hora dos ratings
3. Existiam filmes com algo frases semelhante a ""' no título, por isso foi criada uma expressão regular para as remover.

```
movies <- movies %>% mutate(moviename =gsub("&#[0-9]*;", "", moviename))
```

4. Remover linhas da datatable profile que tinham NA's
5. Foi criada uma tabela chamada aggMovies onde continha o idMovie, o número de Ratings que esse filme teve e a médias das suas avaliações, e com essa tabela:
    - Existia muitos filmes com pouca ratings, então para termos filmes com um número substancial de rating, removemos os filmes que não tinham um número de ratings maior que 2000
    - Removemos esses mesmos filmes da tabela movies
    - Removemos as avaliações que continha esses filmes removidos

Exploratory data

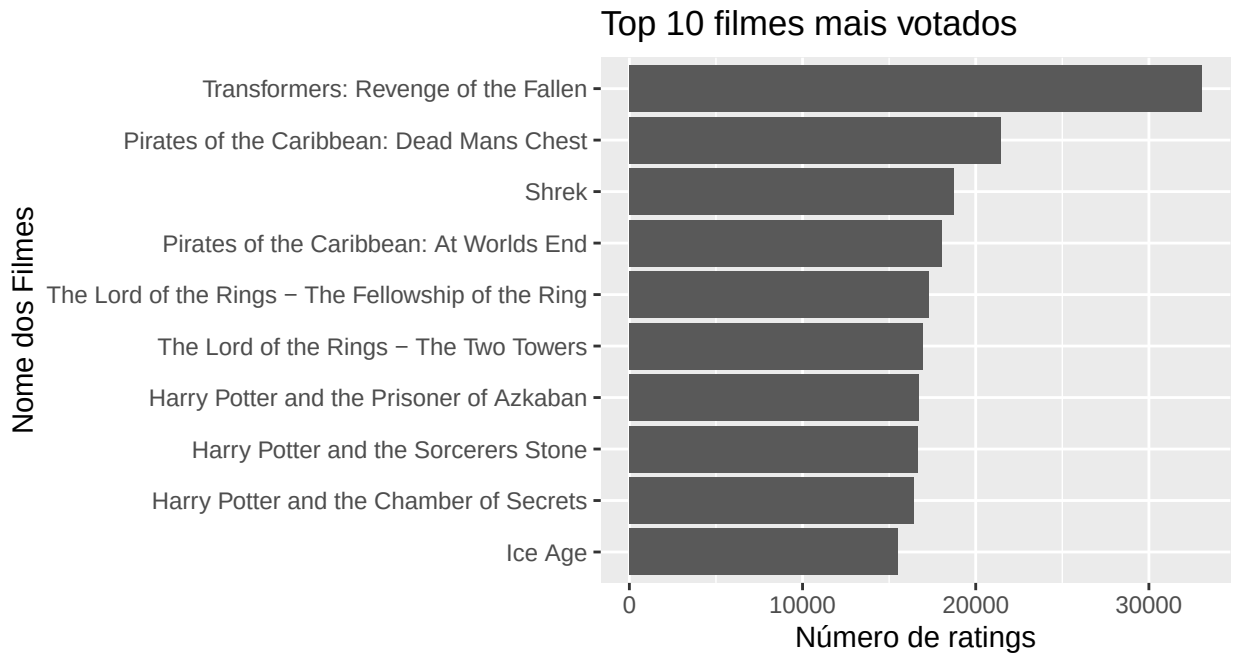
Análise dos filmes

Com o objetivo de fazer uma avaliação geral dos filmes foram considerados a quantidade de reviews e os ratings atribuídos pelos utilizadores.
Assim, foi obtido o seguinte gráfico para considerar os top-10 filmes mais votados:

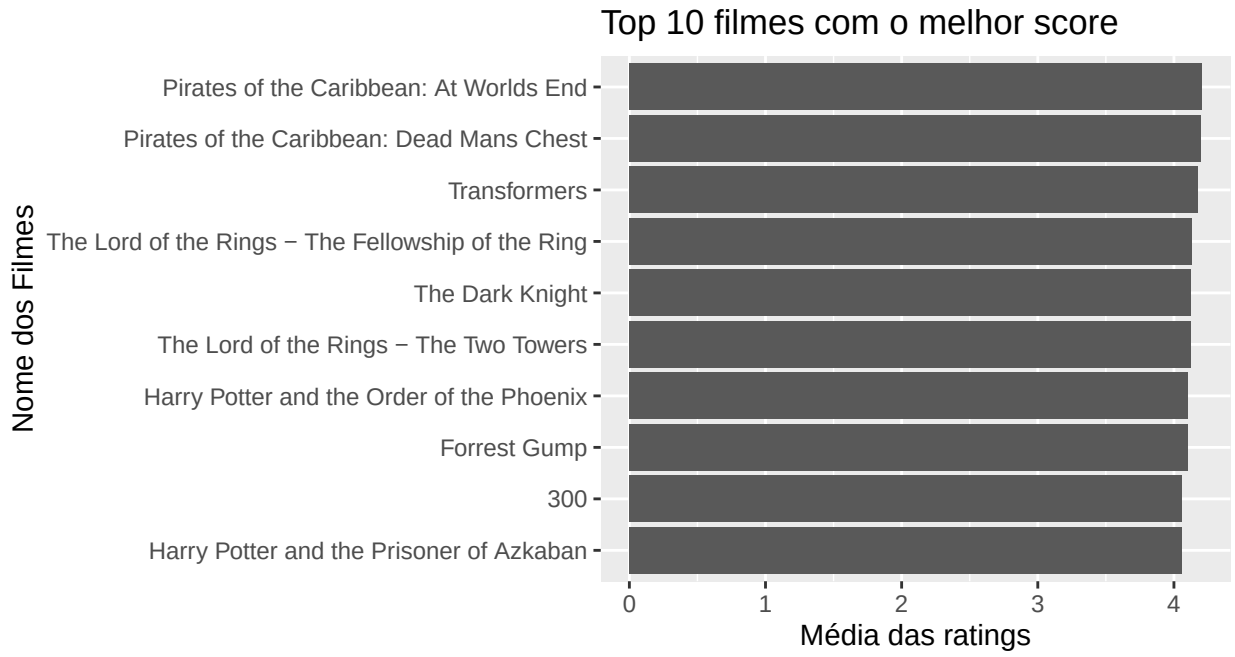
Table 1: Os melhores filmes mais votados e com melhor score

	moviename	numRatings	AvgRating
5	Pirates of the Caribbean: At Worlds End	18026	4.204538
6	Pirates of the Caribbean: Dead Mans Chest	21477	4.200587
13	Transformers	14912	4.177441
11	The Lord of the Rings - The Fellowship of the Ring	17321	4.132123
9	The Dark Knight	10538	4.126352
12	The Lord of the Rings - The Two Towers	16920	4.121543
3	Harry Potter and the Order of the Phoenix	15223	4.102838
2	Forrest Gump	14157	4.101505
1	300	12611	4.059987
4	Harry Potter and the Prisoner of Azkaban	16725	4.056652
7	Saving Private Ryan	10841	4.021539
10	The Lion King	15507	4.009028
8	Shrek	18730	4.000934

De seguida foram analisados os top-10 filmes com o melhor score:

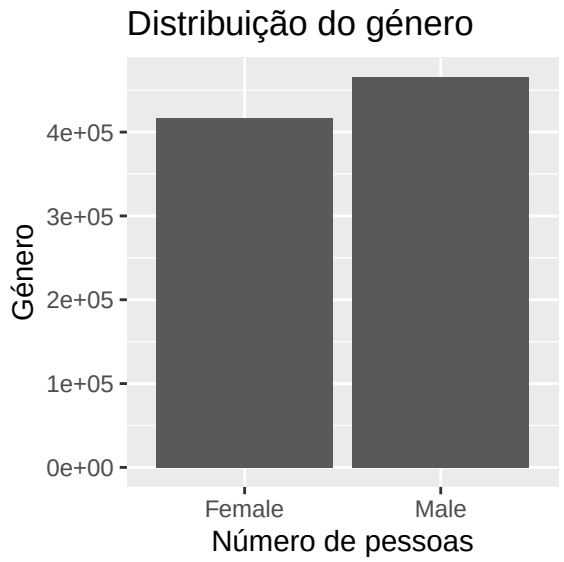


Assim, pode-se afirmar que existe uma correlação positiva entre os filmes mais votados e os filmes com os melhores scores, visto estas duas classes têm seis filmes em comum, como por exemplo 'Pirates of the Caribbean: Dead Mans Chest' que é o segundo filme mais votado e o segundo filme com melhor score.



Análise dos utilizadores

Relativamente a análise de utilizadores foi verificado que o género de utilizadores foi dividido proporcionalmente entre homens e mulheres.



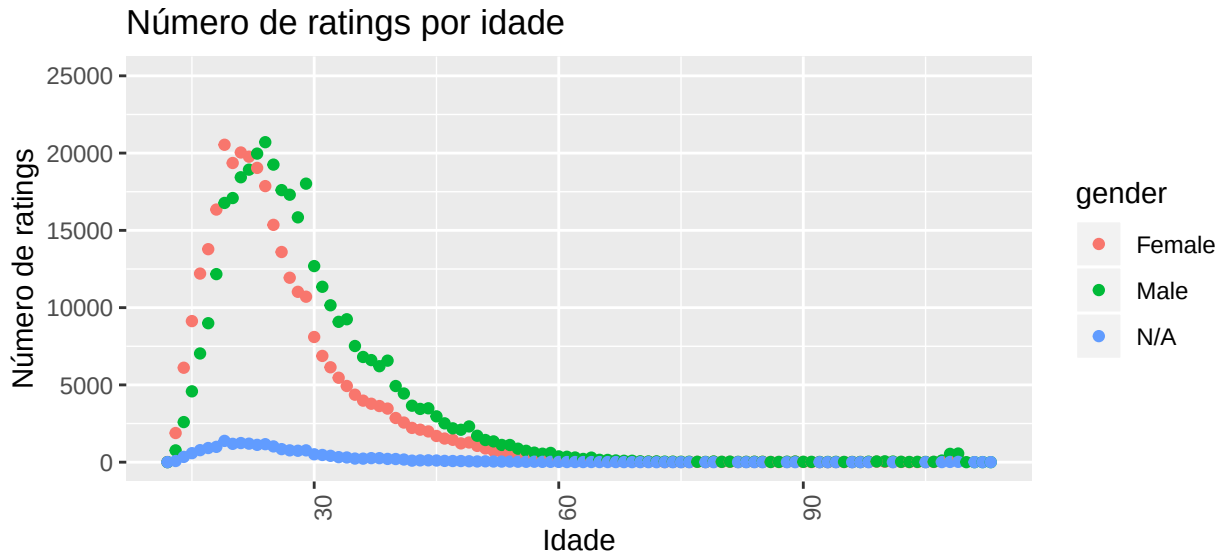
De seguida foi feita uma análise relativamente ao número de votos de acordo com a idade e o género dos utilizadores.

É de notar que, para análise de número de ratings por idade, foi guardada a informação 'NA' sobre o género, isto é, para além dos géneros masculino e feminino, foi guardada a informação dos utilizadores cuja informação de sexo se perdeu, pois como se verificou no gráfico que segue, independentemente do género o número de reviews atinge o seu pico, aproximadamente nos 20 anos de idade.

Assim, foi verificado o comportamento (relativamente a deixar uma avaliação sobre um filme) bastante semelhante entre homens e mulheres, independentemente da idade em geral, exceto no caso de utilizadores mulheres entre 20 e 60 anos, que deixam ligeiramente menos reviews do que os homens na mesma idade.

Relativamente aos utilizadores com o género 'N/A', os mesmos seguem aproximadamente a mesma curva de número de reviews como os homens e mulheres, mas com as quantidades menores.

Verificou-se que independentemente do género da pessoa, a idade mais propícia a reviews é entre 20 e 35 anos.



Modeling approaches

Para testar os modelos foram criados dois utilizadores:

- Um utilizador com o nome de newUserReal, que contém os dados dos seus ratings e que vai ser transformado numa "RealRatingMatrix". Os ratings do utilizador:

```
## # A tibble: 9 x 3
##   moviename                    movieid rating
##   <chr>                          <int>  <dbl>
## 1 Ghost Busters (Ghostbusters)   22375    4.5
## 2 I Am Legend                    26324    3.2
## 3 Mrs. Doubtfire                 36803    4.2
## 4 Rocky Balboa                   42171    3
## 5 Scary Movie 2                  46658    4.6
## 6 Speed                          49266    2.6
## 7 The Craft                      53927    1.5
## 8 The Truman Show                60584    4.7
## 9 Van Helsing                    64804    2
```

- Um outro utilizador com o nome de newUserBinary que viu os mesmo filmes, mas em vez de ter os ratings, tem só o valor de 1 (True) a indicar que ele os viu. Esse utilizador vai ser transformado numa "BinaryRatingMatrix".

Popularity

Informação binária

Para obter um modelo de recomendação baseado na popularidade e com dados binários, primeiro convertemos a tabela ratingsTimed numa "BinaryRatingMatrix".

A seguir criamos o modelo com o método "POPULAR"

```
modelPop <- Recommender(data=popMatrix, method="POPULAR")
```

Usando o utilizador newUserBinary obtemos as seguintes recomendações com valor de N (1,2,5)

```
## # A tibble: 1 x 2
##   moviename                       movieid
##   <chr>                             <dbl>
## 1 Transformers: Revenge of the Fallen   62530
```

```
## # A tibble: 2 x 2
##   moviename                                movieid
##   <chr>                                      <dbl>
## 1 Transformers: Revenge of the Fallen        62530
## 2 Pirates of the Caribbean: Dead Mans Chest   42237
```

```
## # A tibble: 5 x 2
##   moviename                                           movieid
##   <chr>                                                 <dbl>
## 1 Transformers: Revenge of the Fallen                   62530
## 2 Pirates of the Caribbean: Dead Mans Chest             42237
```

```
## 3 Shrek                                                  45119
## 4 Pirates of the Caribbean: At Worlds End                39384
## 5 The Lord of the Rings - The Fellowship of the Ring     56915
```

Informação não binária

Para obter um modelo de recomendação baseado na popularidade e com dados não binários, primeiro convertemos a tabela ratingsTimed numa "RealRatingMatrix".

A seguir criamos o modelo com o método "POPULAR"

```
modelPop <- Recommender(data=popMatrix, method="POPULAR")
```

Usando o utilizador newUserReal obtemos as seguintes recomendações com valor de N (1,2,5)

```
## # A tibble: 1 x 2
##   moviename                                movieid
##   <chr>                                      <dbl>
## 1 Pirates of the Caribbean: Dead Mans Chest   42237
```

```
## # A tibble: 2 x 2
##   moviename                                movieid
##   <chr>                                      <dbl>
## 1 Pirates of the Caribbean: Dead Mans Chest   42237
## 2 Pirates of the Caribbean: At Worlds End     39384
```

```
## # A tibble: 5 x 2
##   moviename                                              movieid
##   <chr>                                                    <dbl>
## 1 Pirates of the Caribbean: Dead Mans Chest                42237
## 2 Pirates of the Caribbean: At Worlds End                  39384
## 3 The Lord of the Rings - The Fellowship of the Ring       56915
## 4 The Lord of the Rings - The Two Towers                   56916
## 5 Forrest Gump                                             17971
```

Association Rules

Informação binária

Para obter um modelo de recomendação baseado em regras de associação e com dados binários, primeiro convertemos a tabela ratingsTimed numa "BinaryRatingMatrix".
A seguir criámos o modelo usando o método "arules":

```
modelAR <- Recommender(assoRulesMatrix,"AR", parameter = list(support=0.05, confidence=0.75))
```

No inicio tínhamos usado um suporte de 0.1 e confiança de 0.75 (o suporte é pequeno porque temos uma matriz muito esparsa), mas como o modelo só gerou 9 regras, não conseguíamos aplicar nenhuma ao nosso utilizador (newUserBinary), então reduzimos o suporte.
Com um suporte de 0.6 obtemos só uma recomendação, mas com 0.5 geramos 16862 regras.

```
## set of 16862 rules
```

E com essas regras conseguimos obter 18 recomendações para o nosso utilizador. Usando um valor de N (1,2,5), obtemos as seguintes recomendações:

```
## # A tibble: 1 x 2
##   moviename                                  movieid
##   <chr>                                        <dbl>
## 1 Pirates of the Caribbean: Dead Mans Chest   42237
```

```
## # A tibble: 2 x 2
##   moviename                                  movieid
##   <chr>                                        <dbl>
## 1 Pirates of the Caribbean: Dead Mans Chest   42237
## 2 Pirates of the Caribbean: At Worlds End     39384
```

```
## # A tibble: 5 x 2
##   moviename                                             movieid
##   <chr>                                                   <dbl>
## 1 Pirates of the Caribbean: Dead Mans Chest             42237
## 2 Pirates of the Caribbean: At Worlds End               39384
## 3 The Lord of the Rings - The Fellowship of the Ring   56915
## 4 The Lord of the Rings - The Two Towers                56916
## 5 Forrest Gump                                          17971
```

Informação não binária

Usando o método "arules" não conseguimos usar uma "RealRatingMatrix".

Collaborative Filtering

Informação binária
O primeiro passo foi converter a tabela ratingsTimed numa "BinaryRatingMatrix".

A seguir foram criados os dois modelos:

- Item Based Collaborative Filtering (IBCF) com o método cosine e k=4
- User Based Collaborative Filtering (UBCF) com o método cosine e nn=3

Com os modelos criados, foi utilizado o utilizador newUserBinary.

Para obter as recomendações para os filmes, aplicamos os dois modelos com diferentes números de N (1,2,5).

Resultados de User Based Collaborative Filtering (UBCF):

```
## # A tibble: 1 x 2
##   moviename         movieid
##   <chr>              <dbl>
## 1 Final Destination   17330
```

```
## # A tibble: 2 x 2
##   moviename                                 movieid
##   <chr>                                       <dbl>
## 1 Final Destination                           17330
## 2 Harry Potter and the Prisoner of Azkaban   20644
```

```
## # A tibble: 5 x 2
##   moviename                                movieid
##   <chr>                                      <dbl>
## 1 Final Destination                          17330
## 2 Harry Potter and the Prisoner of Azkaban   20644
## 3 Monster-in-Law                             34057
## 4 Scary Movie 3                              46659
## 5 Scary Movie 4                              46660
```

Resultados de Item Based Collaborative Filtering (IBCF):

```
## # A tibble: 1 x 2
##   moviename     movieid
##   <chr>           <dbl>
## 1 Scary Movie 3   46659
```

```
## # A tibble: 2 x 2
##   moviename     movieid
##   <chr>           <dbl>
## 1 Scary Movie 3   46659
## 2 Home Alone      25422
```

```
## # A tibble: 5 x 2
##   moviename         movieid
##   <chr>               <dbl>
## 1 Scary Movie 3       46659
## 2 Home Alone          25422
## 3 Jumanji             28315
## 4 Dr. Dolittle        14813
## 5 Miss Congeniality   36096
```

Informação não binária
Este procedimento foi idêntico ao anterior, mas em vez de convertermos a tabela numa "BinaryRatingMatrix", convertemos numa "RealratingMatrix" e usamos o utilizador newUserReal.

Para obter as recomendações para os filmes, aplicamos os dois modelos com diferentes número de N (1,2,5).

Resultados de User Based Collaborative Filtering (UBCF):

```
## # A tibble: 1 x 2
##   moviename    movieid
##   <chr>          <dbl>
## 1 Shes the Man   47246
```

```
## # A tibble: 2 x 2
##   moviename    movieid
##   <chr>          <dbl>
## 1 Shes the Man   47246
## 2 A Bugs Life      926
```

```
## # A tibble: 5 x 2
##   moviename    movieid
##   <chr>          <dbl>
```

```
## 1 Shes the Man   47246
## 2 A Bugs Life      926
## 3 Cinderella     10290
## 4 Mulan          34573
## 5 Shrek          45119
```

Resultados de Item Based Collaborative Filtering (IBCF):

```
## # A tibble: 1 x 2
##   moviename      movieid
##   <chr>            <dbl>
## 1 American Pie 2    3317
```

```
## # A tibble: 2 x 2
##   moviename              movieid
##   <chr>                    <dbl>
## 1 American Pie 2            3317
## 2 Not Another Teen Movie   36907
```

```
## # A tibble: 5 x 2
##   moviename                   movieid
##   <chr>                         <dbl>
## 1 American Pie 2                 3317
## 2 Not Another Teen Movie        36907
## 3 Scary Movie 3                 46659
## 4 Scary Movie 4                 46660
## 5 Back to the Future Part III    5000
```

Analise of the results

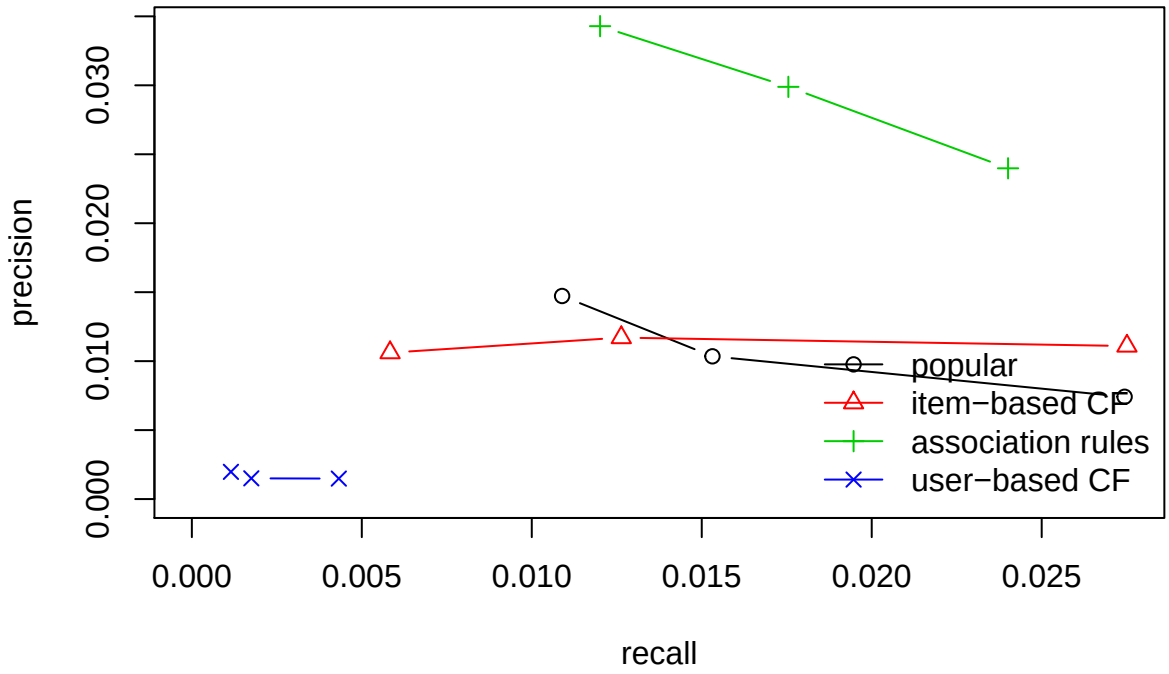
Com o objetivo de avaliar a performance dos modelos foi usado método de validação cruzada. Tendo em conta que o modelo de association rules só foi possivel usando a binaryRatingMatrix, este só foi testado em relação a isso. Os restantes modelos falados neste trabalho, foram restados usando as duas variantes das matrizes.
Seguidamente foi definido o 5-fold cross validation para avaliação dos modelos e os respetivos métodos de previsão.

```
# definicao de 5-fold cross validation
ecross_real <- evaluationScheme(rat_real_matrix, method="cross-validation",
                                k=5, given=-1, goodRating=0)

ecross_binary <- evaluationScheme(rat_binary_matrix, method="cross-validation",
                                  k=5, given=-1, goodRating=0)

### metodo de previsão
methods_real <- list(popular = list(name = "POPULAR", param = NULL),
        `user-based CF` = list(name = "UBCF", param = list(method = "cosine", nn = 3)),
        `item-based CF` = list(name = "IBCF", param = list(method = "cosine", k = 4)))


methods_binary  <- list(AR = list(name ="AR", param = list(support=0.05, confidence=0.75)),
          popular = list(name = "POPULAR", param = NULL),
```

```
        `user-based CF` = list(name = "UBCF",param = list(method = "cosine", nn = 3)),
        `item-based CF` = list(name = "IBCF", param = list(method = "cosine", k = 4)))
```

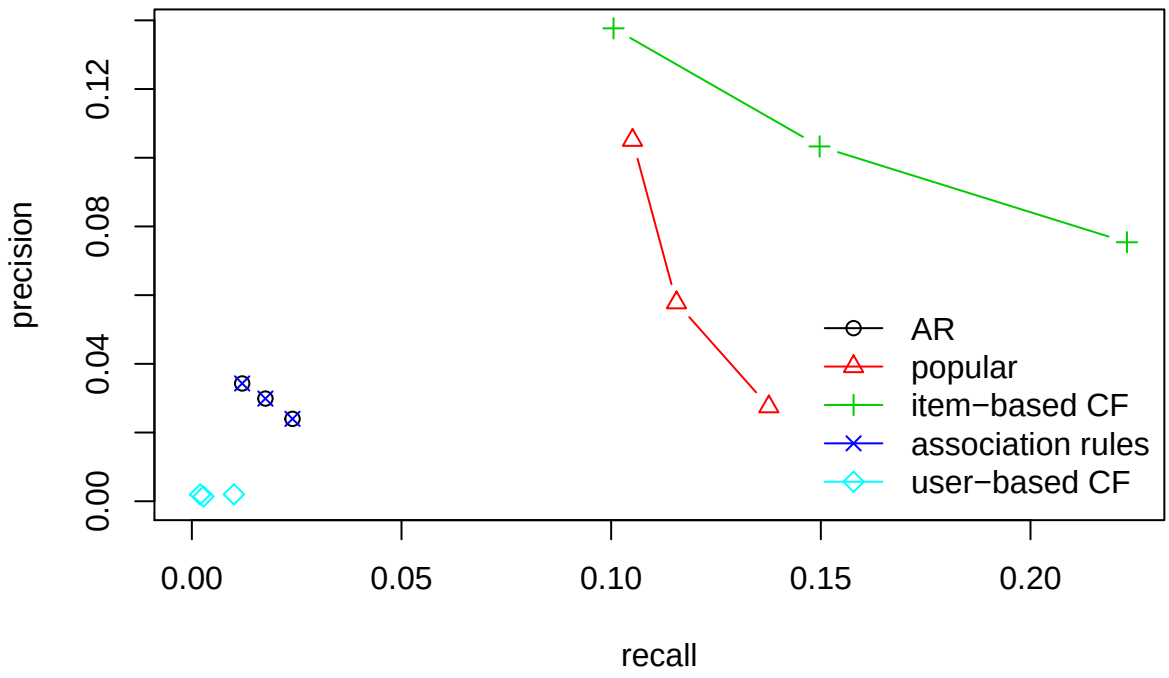
Por fim, foram obtidos os resultados para os métodos de popularidade, IBCF, UBCF e regras de associação. Dado que tínhamos um grande número de dados, houve problemas com o espaço na RAM quando corríamos o método UBCF. Por isso foi feito um sampling de 200000 linhas da table ratingsTimed.
Para que seja possível visualizar as medidas de precisão e do recall de todos os modelos, prosseguiu-se com a junção de resultados das regras de associação com os restantes métodos e foi obtido o seguinte gráfico:

Matrix Binária



Relativamente ao método de binaryRatingMatrix, os resultados indicam que o item-besed CF é o método que apresenta os melhores resultado de precisão e de recall, apesar de diminuir com o número de recomendações. A performance do método de popularidade tem um comportamento semelhante ao de item-based CF, com a diferença de que decresce muito mais rapidamente com o aumento de número de recomendações O método de regras de associação e o user-based CF são os que têm a pior performance, sendo o método de associação é ligeiramente melhor do que o de user-based CF, tanto na precisão como no recall.

Matrix Real



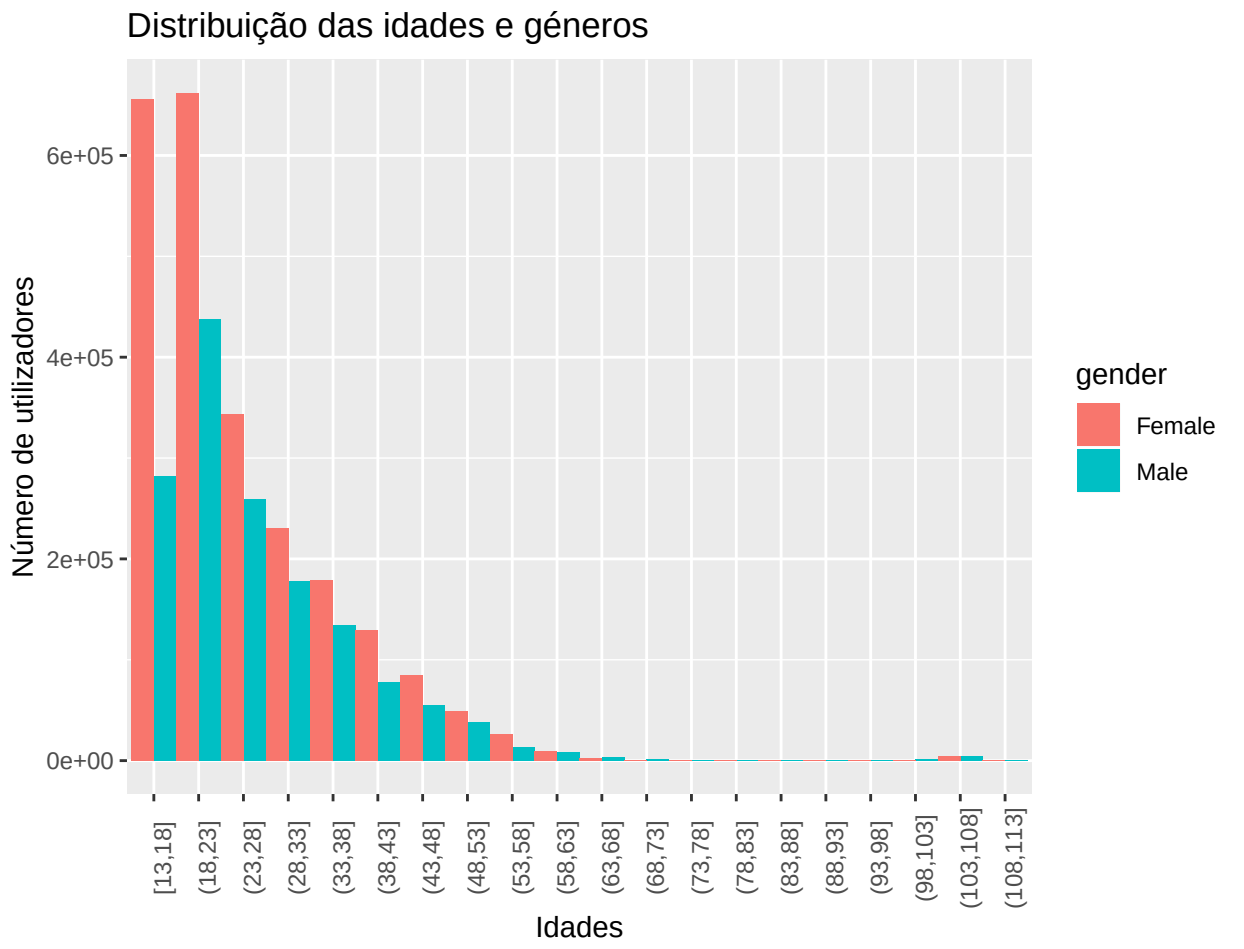
Relativamente ao método de realRatingMatrix, os resultados indicam que item-based-CF e o método de popularidade são métodos que tem o melhor recall, porém o item-based-CF ganha na precisão para 5 filmes recomendados. Por sua vez, o método de popularidade, para um filme recomendado é o método que tem a maior precisão. Independentemente de número de recomendações o user-based-CF é o método com os piores resultados de precisão e de recall.

Em geral, o método de binaryRatingMatrix tem uma performance muito melhor do que o realRatingMatrix.

Context-aware recommendations

Para abordar o problema de recomendação de acordo com um contexto, foi adotada a seguinte estratégia:

- A informação relativa as idades e do género dos utilizadores foi distribuída em ranges de 5 anos, como por exemplo, consegue-se visualizar na seguinte tabela, de acordo com o número de pessoas que satisfazem estas condições:



Foi verificado que na sample a usar, havia mais utilizadores entre 15 e 20 anos tanto do sexo masculino como o de feminino; e com o avançar da idade o número de utilizadores tende a diminuir.

Após a divisão da informação de acordo com os ranges das idades e do género dos utilizadores é aplicado o método da popularidade, de acordo com as seguintes instruções:

- Como exemplo foi escolhido um utilizador aleatório do sexo feminino com 28 anos de idade e de seguida foi encontrado o range das idades que contém este valor:

```
#select user with age 28 and gender female
x <- 28
g <- "Female"

#find range that contains age value
for(i in age_range){
  if ( (x >= as.numeric( sub("\\((.+),.*", "\\1", i)) ) &
       (x < as.numeric( sub("[^,]*,([^]]*)\\]", "\\1", i) ))){

    selected_range <- c(as.numeric( sub("\\((.+),.*", "\\1", i)),
    as.numeric( sub("[^,]*,([^]]*)\\]", "\\1", i) ))
    }
  }
```

- Assim obtém-se a informação de utilizadores que têm as mesmas características que o utilizador escolhido aleatoriamente para fazer a recomendação (nesta tabela só mostramos as 10 primeiras linhas) :

Table 2: Utilizadores com as mesmas características

userid	gender	location	memberfor	lastlogin	profileview	age
611822	Female	221	2009-06-03 00:00:00	23	32	32
781852	Female	264	2009-11-01 00:00:00	91	29	29
868001	Female	77	2009-09-01 00:00:00	3	29	29
1011935	Female	255	2009-10-03 00:00:00	26	29	29
204758	Female	197	2009-10-02 00:00:00	153	31	31
224170	Female	485	2009-10-02 00:00:00	71	30	30
1008962	Female	714	2009-02-01 00:00:00	98	31	31
546888	Female	304	2009-11-01 00:00:00	23	29	29
235450	Female	345	2009-10-02 00:00:00	278	31	31
960176	Female	338	2009-06-01 00:00:00	16	29	29

- De seguida é aplicado o método de popularidade a informação previamente filtrada por range das idades e o género; e é obtido o resultado da recomendação (usando o utilizador newUserReal)

```
## # A tibble: 5 x 2
##   moviename                                  movieid
##   <chr>                                        <dbl>
## 1 Pirates of the Caribbean: Dead Mans Chest   42237
## 2 Shrek                                       45119
## 3 The Green Mile                              55562
## 4 The Lord of the Rings - The Two Towers      56916
## 5 The Lion King                               54612
```

Conclusions, shortcomings and future work

Visto que para cada modelo (excepto association rules) foram aplicadas as duas abordagens: Usando as ratings dos utilizadores (realRantingMatrix) ou a indicação se um utilizador viu um determinado filme (binaryRatingMatrix), era esperado ter recomendações diferentes para o mesmo modelo mas com matrizes diferentes, o que se verificou.
No âmbito dos sistemas de recomendação sensíveis ao contexto, seria uma mais valia desenvolver um algoritmo de web scrapping que possa acrescentar a informação do género dos filmes (comédia, thriller, drama, etc). Assim, para além de fazer um sistema de recomedação baseado no range das idades e do sexo do utilizador, seria possível recomendar um filme com maior precisão, ao dispistar os géneros de filmes preferidos do utilizador em questão.
Como trabalho futuro seria aplicar os modelos a utilizadores reais e testar as suas precisões.