Movies Recommendation in Flixster

Robson Teixeira and Nuno Gomes

17/05/2020

Contents

1 Abstract **2**

2 Introduction **3**

2.1 Flixster Dataset 3

3 Methods and Analysis **6**

3.1 Data Preparation 6

3.2 Data Exploration 7

Chapter 1

Abstract

We report on the methodology we adopted to create a recommendation system for films using the Flixster data set, and on the results we obtained. Our model is based on the assumption that if a group of users had liked the same films in the past, they will like similar movies in the future. Hence, if two users have a similar rating history (movies and ratings) and one of them has recently enjoyed a film that the other has not seen yet, then that movie is proposed to the latter. The recommendation system uniquely takes into account the user ratings, and does not consider the characteristics of the films. ADD MAIN RESULTS HERE.

Chapter 2

Introduction

Recommender or **recommendation** systems is a branch of *Web usage mining* aiming at predicting the “preferences” or the “rating” a user would give to an item. They are widely used on the Web, mainly in online streaming services, such as YouTube, Amazon, or Netflix (just to name a few), and e-commerce applications (eBay, Amazon, OLX, *etc.*), in order to recommend products and services to the users. They serve three important functions: (*i*) to increase the profit of companies, (*ii*) to help users to select specific products within the available offer, by giving them personalised recommendations based on previous interactions, and (*iii*) to predict the rating for a new item. Companies struggle for customer loyalty on a daily basis, and while doing so, they invest on recommender systems that try to increase the likelihood of purchase, by analysing customers’ preferences and past interactions.

The importance of recommender systems on the success of businesses and on company-customer relationships can be inferred from the one million dollar prize that Netflix offered in 2006 to the person or team that could improve their recommendation system by at least 10%.[1] The success of companies such as Amazon or platforms like Youtube lies partly in the recommendation and marketing strategies, based on the user preferences.

During this project, we created a film recommendation system using the Flixster data set, while applying the knowledge acquired in the course of Advanced Topics in Data Science/Data Mining II over the second semestre of the Masters in Data Science.

This report is organised as follows:

2.1 Flixster Dataset

The `Flixster` is a social movie site which allow users to share movie ratings, discover new movies and meet others with similar movie taste. The dataset generated from `Flisxter` site consist of 8196077 ratings of 48794 movies by 147612 users. The data are distributed in three files:

[1] https://www.netflixprize.com/

- `profile.txt` : contains informations regarding the users like user id, gender, age, location, for how long it has been a member
- `movie-names.txt` : contains information regarding movies as name and movie id
- `Ratings.time.txt` : contains information on the ratings given by users to movies and on which date

A summary of *movies* is given below, togeher with several first rows of dataframe:

```
## Rows: 66,730
## Columns: 2
## $ moviename <chr> "$ (Dollars) (The Heist)", "$5 a Day (Five Dollars a Day)...
## $ movieid   <int> 252, 253, 1, 254, 255, 256, 257, 2, 258, 3, 4, 5, 259, 26...
```

```
##    variable q_zeros p_zeros q_na p_na q_inf p_inf      type unique
## 1 moviename       0       0    0    0     0     0 character  66730
## 2   movieid       0       0    0    0     0     0   integer  66730
```

```
## # A tibble: 6 x 2
##   moviename                                        movieid
##   <chr>                                              <int>
## 1 $ (Dollars) (The Heist)                              252
## 2 $5 a Day (Five Dollars a Day)                        253
## 3 $9.99                                                  1
## 4 $windle (Swindle)                                    254
## 5 &#034;BBC2 Playhouse&#034; Caught on a Train        255
## 6 &#034;Independent Lens&#034;  Race to Execution     256
```

A summary of *users* is given below, togeher with several first rows of dataframe:

```
## Rows: 1,002,796
## Columns: 7
## $ userid      <int> 981904, 882359, 921220, 798641, 952904, 888, 300293, 80...
## $ gender      <chr> "Male", "Female", "Female", "Male", "Female", "Female",...
## $ location    <int> 111, 870, 993, 250, 172, 157, 221, 180, 233, 221, 282, ...
## $ memberfor   <chr> "2009-09-01 00:00:00", "2009-10-02 00:00:00", "2009-09-...
## $ lastlogin   <int> 1, 181, 124, 40, 44, 39, 91, 1, 26, 23, 490, 1097, 272,...
## $ profileview <chr> "19", "108", NA, "22", "21", "18", "13", NA, NA, "32", ...
## $ age         <chr> "19", "108", NA, "22", "21", "18", "13", NA, NA, "32", ...
```

```
##      variable q_zeros p_zeros   q_na  p_na q_inf p_inf      type  unique
## 1      userid       0    0.00      0  0.00     0     0   integer 1002796
## 2      gender       0    0.00  67529  6.73     0     0 character       2
## 3    location   57726    5.76    203  0.02     0     0   integer    1429
## 4   memberfor       0    0.00    203  0.02     0     0 character      36
## 5   lastlogin   48949    4.88  57925  5.78     0     0   integer    3099
## 6 profileview       0    0.00 255564 25.49     0     0 character     126
## 7         age       0    0.00 255564 25.49     0     0 character     126
## # A tibble: 6 x 7
##   userid gender location memberfor           lastlogin profileview age
##    <int> <chr>     <int> <chr>                   <int> <chr>       <chr>
## 1 981904 Male        111 2009-09-01 00:00:00         1 19          19
## 2 882359 Female      870 2009-10-02 00:00:00       181 108         108
## 3 921220 Female      993 2009-09-01 00:00:00       124 <NA>        <NA>
## 4 798641 Male        250 2009-11-01 00:00:00        40 22          22
## 5 952904 Female      172 2009-11-01 00:00:00        44 21          21
## 6    888 Female      157 2009-10-01 00:00:00        39 18          18
```

The variables *gender* and *age* are presented as character and should be changed to factor and integer respectively.

The variables *location*, *memberfor*, *lastlogin* and *profileview* are not needed for the context this analysis, so they should be removed.

There are a significant number of *NA's* in variables *gender* (67529 or around 6.73% of total) and *age* (255618 or around 25.49% of total). We adopt the aproach the replacement the empty values into a new value to treatmet of missing values. The empty values into variable *gender* will be replaced for new value `Other`. For the empty values into *age* variable will be set the value `0`.

A summary of *ratings* is given below, togeher with several first rows of dataframe:

```
## Rows: 8,196,077
## Columns: 4
## $ userid  <int> 882359, 882359, 882359, 882359, 882359, 882359, 882359, 882...
## $ movieid <int> 81, 926, 1349, 2270, 3065, 3522, 3583, 4216, 4871, 4917, 51...
## $ rating  <dbl> 1.5, 1.0, 2.0, 1.0, 5.0, 0.5, 0.5, 0.5, 3.5, 0.5, 1.0, 1.0,...
## $ date    <chr> "2007-10-10 00:00:00", "2007-10-10 00:00:00", "2007-10-10 0...
```

```
##   variable q_zeros p_zeros q_na p_na q_inf p_inf      type unique
## 1   userid       0       0    0    0     0     0   integer 147612
## 2  movieid       0       0    0    0     0     0   integer  48794
## 3   rating       0       0    0    0     0     0   numeric     10
## 4     date       0       0    0    0     0     0 character   1450
```

```
## # A tibble: 6 x 4
##   userid movieid rating date
##    <int>   <int>  <dbl> <chr>
## 1 882359      81    1.5 2007-10-10 00:00:00
## 2 882359     926    1   2007-10-10 00:00:00
## 3 882359    1349    2   2007-10-10 00:00:00
## 4 882359    2270    1   2007-10-10 00:00:00
## 5 882359    3065    5   2007-12-29 00:00:00
## 6 882359    3522    0.5 2007-11-13 00:00:00
```

Chapter 3

Methods and Analysis

3.1 Data Preparation

In this section we prepare the dataset to be used in the analysis. Some steps of cleaning data are applied.

```r
# Remove no relevant variables
users <- select(users, -c(location, memberfor, lastlogin, profileview) )

# Setting 'gender' as factors and 'age' as integer
users <- users %>% mutate(gender = as.factor(gender)) %>% mutate(age = as.integer(age))

# Add column range_age into user data set
users <- users %>% mutate(range_age = cut_width(age, 10, boundary = 0) )

# replace NA values into gender with "unknown"
users <- users %>% mutate(gender = replace(gender, is.na(gender), "unknown"))

# replace NA values into age with "0"
users <- users %>% mutate(age = replace(age, is.na(age), 0))

flixster <- ratings %>% left_join(users, by = "userid") %>% left_join(movies, by = "movi

remove(movies, users, ratings)
```

We split the dataset in two parts, the training set called `train_set` and the test set called `test_set` with 70% and 30% of the original dataset respectively.

```r
# 'test_set' will be 30% of dataset
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(y = flixster$rating, times = 1, p = 0.3, list = FALSE)
```

```r
train_set <- flixster[-test_index,]
temp <- flixster[test_index,]

# Make sure userid and movieid in 'test_set' are also in 'train_set'
test_set <- temp %>% semi_join(train_set, by = "movieid") %>% semi_join(train_set, by =

# Add rows removed from 'test_set' set back into 'train_set' set
removed <- anti_join(temp, test_set)
```

```
## Joining, by = c("userid", "movieid", "rating", "date", "gender", "age", "range_age",
```

```r
train_set <- rbind(train_set, removed)

rm(test_index, temp, flixster, removed)
```

3.2 Data Exploration

Before start building the model, we need to understand the structure of the data, the distribution of variables and the relationship of the predictors. This information will help build a better model.

3.2.1 Profile Data Set

```r
glimpse(train_set)
```

```
## Rows: 5,757,706
## Columns: 8
## $ userid    <int> 882359, 882359, 882359, 882359, 882359, 882359, 882359, 8...
## $ movieid   <int> 1349, 2270, 3522, 3583, 4216, 4871, 5124, 9180, 10033, 11...
## $ rating    <dbl> 2.0, 1.0, 0.5, 0.5, 0.5, 3.5, 1.0, 1.0, 1.0, 2.0, 5.0, 2....
## $ date      <chr> "2007-10-10 00:00:00", "2007-10-10 00:00:00", "2007-11-13...
## $ gender    <fct> Female, Female, Female, Female, Female, Female, Female, F...
## $ age       <dbl> 108, 108, 108, 108, 108, 108, 108, 108, 108, 108, 108, 10...
## $ range_age <fct> "(100,110]", "(100,110]", "(100,110]", "(100,110]", "(100...
## $ moviename <chr> "A Nightmare on Elm Street 3 - Dream Warriors", "Aladdin"...
```

From this initial exploration, we can see that `train_set` consists of 5757622 observations spread across 8 variables. The user information are stored in userid, gender, age, range_age columns; the movie information are stored in movieid and moviename columns; the rating information are stored in rating and date columns.

Variables	Type	Description
userid	integer	User unique identifier
movieid	integer	Movie unique identifier

Variables	Type	Description
rating	numeric	rating movie set by user
date	factor	Date that user rating movie
gender	factor	User gender
age	integer	Age of user
range_age	factor	rande age of user
moviename	character	movie name