Movie recommendation task (MOV)

Predict the user’s rating for a given movie

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter</td>
<td>?</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Paul</td>
<td>2</td>
<td>5</td>
<td>?</td>
</tr>
<tr>
<td>Mary</td>
<td>2</td>
<td>4</td>
<td>?</td>
</tr>
</tbody>
</table>

E.g., predict Mary’s rating for the Some Like it Hot movie
MOV – Available data

• About users

<table>
<thead>
<tr>
<th></th>
<th>age</th>
<th>gender</th>
<th>occupation</th>
<th>zip code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter</td>
<td>19</td>
<td>M</td>
<td>student</td>
<td>58644</td>
</tr>
<tr>
<td>Mary</td>
<td>50</td>
<td>F</td>
<td>healthcare</td>
<td>60657</td>
</tr>
</tbody>
</table>

• About movies

<table>
<thead>
<tr>
<th>title</th>
<th>action</th>
<th>...</th>
<th>IMDb rating</th>
<th>director</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toy Story</td>
<td>0</td>
<td>...</td>
<td>8.3</td>
<td>John Lasseter</td>
</tr>
<tr>
<td>Some Like It Hot</td>
<td>0</td>
<td>...</td>
<td>8.3</td>
<td>Billy Wilder</td>
</tr>
<tr>
<td>Star Wars</td>
<td>1</td>
<td>...</td>
<td>8.7</td>
<td>George Lucas</td>
</tr>
</tbody>
</table>
MOV – Getting examples

- Create a database of movies to be rated by users
- Set up a rating scale allowing users to rate movies
- Record users’ ratings
- Typically, the dataset of ratings is sparse.
  So do some pruning, like require a minimum of twenty ratings per user
### MOV – Data

#### Basic statistics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>number of ratings</td>
<td>100,000</td>
</tr>
<tr>
<td>number of movies</td>
<td>1,682</td>
</tr>
<tr>
<td>number of users</td>
<td>943</td>
</tr>
</tbody>
</table>
## MOV – Data representation

<table>
<thead>
<tr>
<th>vote id</th>
<th>MOVIE</th>
<th>USER</th>
<th>RATING</th>
<th>TIMESTAMP</th>
<th>user features</th>
<th>movie features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1997-09-23 00:02:38</td>
<td>24 M technician 85711</td>
<td>Toy Story (1995)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>100,000</td>
<td>1682</td>
<td>916</td>
<td>3</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

See the feature description mov.pdf at [https://ufal.mff.cuni.cz/course/npfl054/materials](https://ufal.mff.cuni.cz/course/npfl054/materials)
Download `mov.development.csv` from
https://ufal.mff.cuni.cz/course/npfl054/materials

```r
> examples <- read.csv("mov.development.csv", sep="\t")

> nrow(examples)
[1] 100000

> str(examples)
'data.frame': 100000 obs. of 33 variables:
$ movie     : int  1 1 1 1 1 1 1 1 1 1 ...
$ user      : int  1 117 429 919 457 468 17 892 16 580 ...
$ rating    : int  5 4 3 4 4 5 4 5 5 3 ...
$ timestamp : int  874965758 880126083 882385785 875289321 ...
$ age       : int  24 20 27 25 33 28 30 36 21 16 ...
$ gender    : Factor w/ 2 levels "F","M": 2 2 2 2 1 2 2 ...
$ occupation: Factor w/ 21 levels "administrator",...
  : 20 19 ...
```

Exercises

• Get the number of different users

```r
> length(unique(examples$user))
[1] 943
```

```r
> ?unique()
... 
Description:

'unique' returns a vector, data frame or array like 'x'
but with duplicate elements/rows removed.
``` ...

• Get the number of different movies

```r
> length(unique(examples$movie))
[1] 1682
```
Exercises

• Get the number of ratings for each user

```
> table(examples$user)
1  2  3  4  5  6  7  8  9 10 11 12 13  ...
272 62 54 24 175 211 403 59 22 184 181 51 636  ...
...
```

• Get the number of ratings for each movie

```
> table(examples$movie)
1  2  3  4  5  6  7  8  9 10 11  ...
452 131 90 209 86 26 392 219 299 89 236  ...
...
```
Exercises

- How many movies has the user(s) with the **highest** number of ratings seen?

```r
> t <- table(examples$user)
> max(t)
[1] 737

# Which user(s)?
> names(which(t == max(t)))
[1] "405"
> examples[examples$user == 405, ][1,]

 movie user rating timestamp age gender occupation zip ...
 ... 405 ... ... 22 F healthcare 10019 ...
Exercises

- How many movies has the user(s) with the **lowest** number of ratings seen?

```r
> min(t)
[1] 20
# Which user(s)?
> names(which(t == min(t)))
[1] "19" "34" "36" "93" "143" "147" "166" "202" "242" ...
[13] "418" "441" "475" "558" "571" "572" "596" "631" "636" ...
[25] "809" "812" "824" "866" "873" "888" "895" "926"
```
• Make a plot of rating vs. number of ratings

```r
> r <- table(examples$rating)
> pdf("ratings.pdf")
> plot(r, main = "MovieLens: frequency",
       xlab = "Rating (1-5)",
       ylab = "Number of ratings",
       type = "p",
       col = "red",
       pch = "O")
> dev.off()
```
MovieLens: frequency

Rating (1−5)

Number of votes

1 2 3 4 5
Exercises

• Get the movies rated at least 3 times
  1. Sort them according to their average rating in descending order
  2. Which of them has the least variance?

```r
# get the movies rated at least 3 times
> m <- table(examples$movie)
> m.1 <- which(m >= 3)

# get average rating of each movie rated at least 3 times
> m.3 <- subset(examples, examples$movie %in% m.1)
> mean <- tapply(m.3$rating, m.3$movie, mean)

# sort them according to their average rating in descending order
# focus on the Top 5
> sort(mean, decreasing=TRUE)[1:5]
   1189  1293  1449  119  408
  5.000000 5.000000 4.625000 4.500000 4.491071
```
# compute standard deviations
> sd <- tapply(m.3$rating, m.3$movie, sd)
sd[[1189]]
# [1] 1.159995
sd[[1293]]
# [1] 0.8164966
sd[[1449]]
# [1] 1.902379
sd[[119]]
# [1] 1
sd[[408]]
# [1] 0.7710474
> movies <- unique(examples[,c(1,9:33)])
> movies[408,]
... 408 Close Shave, A (1995) ... 0 chloupek