Assignment01

Mohamed Elmoudni

February 12, 2017

Project 1 :

Requirements:

The goal of this assignment is to help you build your intuition about recommender systems, with a basic soup to nuts implementation coded "from scratch."

Your task is to build a very basic recommender system, first by writing your own functions, then by replacing those functions with those provided in an R Package or a Python library (such as scikitlearn).

- You should very briefly first describe the recommender system that you're going to build out from a business perspective, e.g. "This system recommends movies to users."
- You can find a dataset, or build out your own toy dataset and load into (for example) an R or pandas dataframe, a Python dictionary or list of lists, (or other data structure of your choosing).
- You can use either collaborative filtering, or a hybrid of content management and collaborative filtering.
- You are encouraged to hand code at least your similarity function.
- After you have built out your own code base, create an alternate version using packages or libraries. Compare the results and performance.
- You are also encouraged to think about how to best handle missing data.
- Your code should be turned in an RMarkdown file or a Jupyter notebook, and posted to Github. You may work in a small group (2 or 3 people) on this assignment. While you're never discouraged from adding features or advanced capabilities such as regularization and matrix factorization methods, it is not expected at this point in the course.

```r
library(recommenderlab)
```

```
## Loading required package: Matrix

## Loading required package: arules

##
## Attaching package: 'arules'

## The following objects are masked from 'package:base':
##
##     abbreviate, write

## Loading required package: proxy

##
## Attaching package: 'proxy'

## The following object is masked from 'package:Matrix':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix
```

```
## Loading required package: registry
```

```
library(reshape2)
library(ggplot2)
library(knitr)
######################################

# Read training file along with header
dt<-read.csv("https://raw.githubusercontent.com/simonnyc/IS-643/master/Assignment01_data.csv", header=TI

#############
# Data exploration


# Just look at first few lines of this file
kable(head(dt))
```

User	Items	Ratings
1	item_1	1
1	item_10	3
1	item_11	1
1	item_12	4
1	item_2	2
1	item_3	3

Data Exploration

```
# Summary
kable(summary(dt), caption ="Data Summary")
```

Table 2: Data Summary

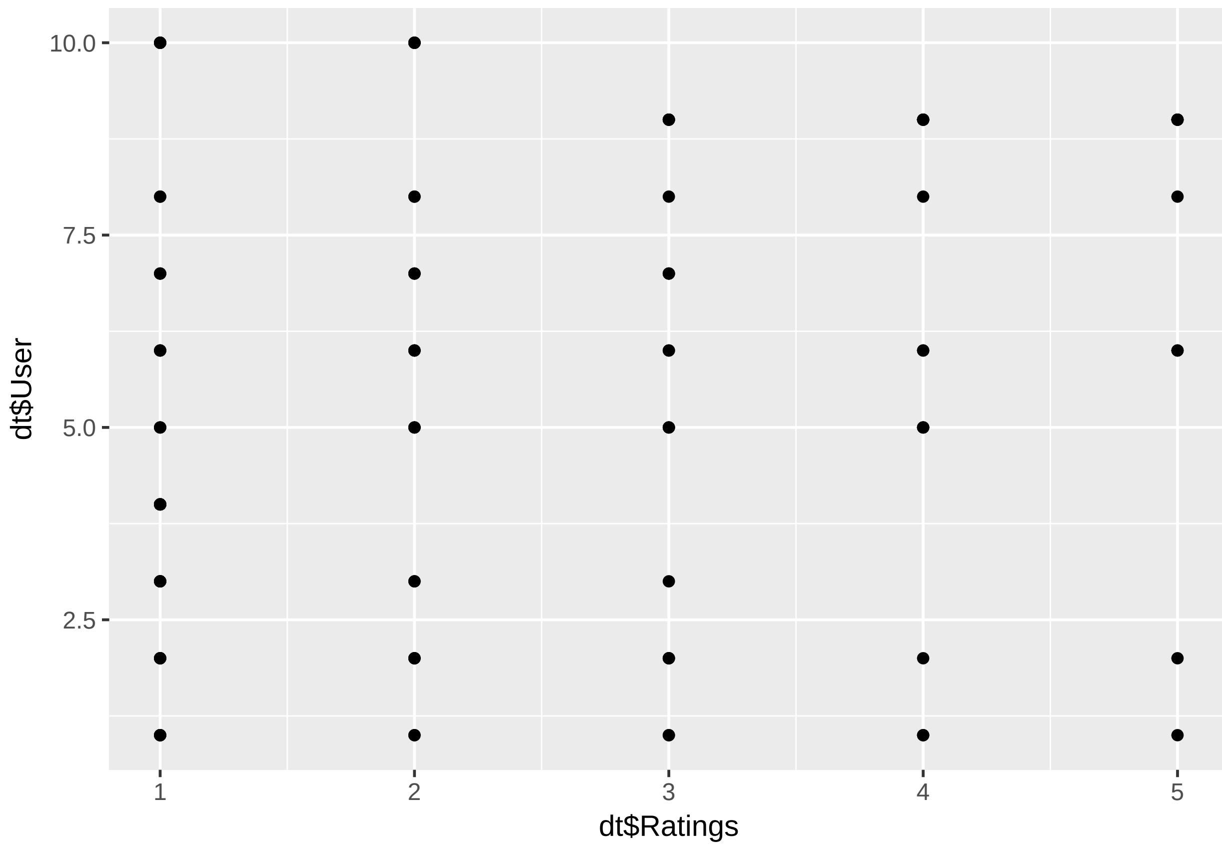
User	Items	Ratings
Min. : 1.00	item_11:10	Min. :1.000
1st Qu.: 3.00	item_12:10	1st Qu.:1.000
Median : 5.00	item_3 :10	Median :2.000
Mean : 5.39	item_1 : 9	Mean :2.295
3rd Qu.: 8.00	item_5 : 9	3rd Qu.:3.000
Max. :10.00	item_9 : 9	Max. :5.000
NA	(Other):48	NA

```
#frequency table

table(dt$Ratings)
```

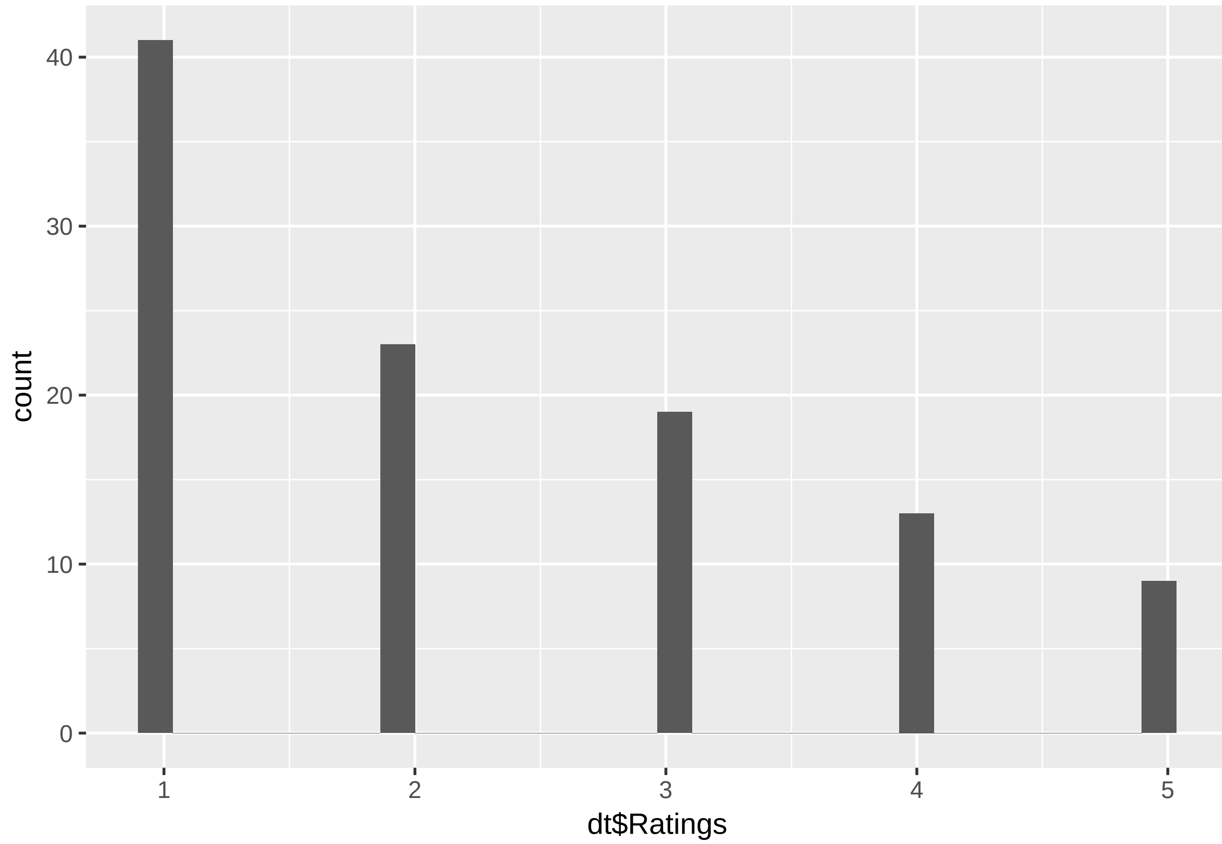
```
##
##  1  2  3  4  5
## 41 23 19 13  9
```

```r
qplot(dt$Ratings, dt$User)
```




```r
qplot(dt$Ratings)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#[head(sort(dt$Ratings,decreasing=TRUE), n = 5)]

#head(dt[sort(dt$Ratings, decreasing=TRUE), ], 100)
```

Implementing Item based recommender systems, like user based collaborative filtering, requires two steps:

1. Calculating Item similarities
2. Predicting the targeted item rating for the targeted User.

Step1: Calculating Item Similarity: we calculate the similarity between co-rated items. We use cosine similarity or pearson-similarity to compute the similarity between items. The output for step is similarity matrix between Items.

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:arules':
##
##     intersect, recode, setdiff, setequal, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: SnowballC
```

```
##
## Attaching package: 'lsa'
## The following object is masked from 'package:dplyr':
##
##     query
```

	item_1	item_2	item_3	item_4	item_5	item_6	item_7	item_8	item_9
item_1	1.0000000	0.7779466	0.8606630	0.6000469	0.6707816	0.6222222	0.8012336	0.5499719	0.6880625
item_2	0.7779466	1.0000000	0.7909058	0.8861991	0.7204843	0.7584980	0.8091134	0.8045086	0.8172515
item_3	0.8606630	0.7909058	1.0000000	0.8133901	0.8785713	0.6196773	0.8235321	0.7485542	0.7804217
item_4	0.6000469	0.8861991	0.8133901	1.0000000	0.8521116	0.7425580	0.8425167	0.9546687	0.8294258
item_5	0.6707816	0.7204843	0.8785713	0.8521116	1.0000000	0.5927270	0.7610782	0.7761505	0.8415606
item_6	0.6222222	0.7584980	0.6196773	0.7425580	0.5927270	1.0000000	0.7765803	0.7919596	0.8256750
item_7	0.8012336	0.8091134	0.8235321	0.8425167	0.7610782	0.7765803	1.0000000	0.9021342	0.8178608
item_8	0.5499719	0.8045086	0.7485542	0.9546687	0.7761505	0.7919596	0.9021342	1.0000000	0.8340577
item_9	0.6880625	0.8172515	0.7804217	0.8294258	0.8415606	0.8256750	0.8178608	0.8340577	1.0000000
item_10	0.6432675	0.8444719	0.7750911	0.9046656	0.8590614	0.7504788	0.8771840	0.9097177	0.9168313
item_11	0.6552976	0.7097184	0.7042830	0.7796603	0.6797220	0.9141401	0.8996469	0.8757605	0.8043478
item_12	0.6913580	0.7779466	0.8606630	0.8375654	0.7927419	0.8000000	0.8012336	0.8328147	0.8027395

Step2: Predicting the targeted item rating for the targeted User

Recommending Top N items: Once all the non rated movies are predicted we recommend top N movies to a user Code for Item based collaborative filtering in R:

```
ratings<- as.matrix(x)
userRecmd <- function(userno)
{

  userRatings <- ratings[userno,]
  userRatings[userRatings==0] <- NA
  #

  non_rated_items <- names(userRatings[is.na(userRatings)])
  rated_items <- names(userRatings[!is.na(userRatings)])
  m1 <- itemSimil[non_rated_items,]
  v1 <- apply(m1,1,function(x) sum((x*ratings[userno,]),na.rm = T)/(sum(x[rated_items])))
  #ratings[userno,names(v1)] <- v1
  #

  return(v1)
}
```

Recommend for User 10

Now we will use the above to find a rating for user 10. Following line predicts the ratings for

```
userRecmd(10)
```

```
##  item_7 item_10
## 1.49148 1.51390
```

Following lines predict the ratings for the user 10 but using built-in package :- RecommenderLab package.

```r
m <- ratings
m[m==0] <- NA

affinity.matrix<- as(m,"realRatingMatrix")
Rec.model<-Recommender(affinity.matrix,method="IBCF",
                       param=list(normalize = "Z-score",method="Cosine", n=5))
```

```
## Warning: Unknown parameters: n

## Available parameter (with default values):
## k     =  30
## method    =  Cosine
## normalize     =  center
## normalize_sim_matrix  =  FALSE
## alpha     =  0.5
## na_as_zero    =  FALSE
## verbose   =  FALSE
```

```r
u <- predict(Rec.model, affinity.matrix[10,], type="ratings")
as(u, "list")
```

```
## $`10`
##   item_7  item_10
## 1.214661 1.483787
```

The numbers are slightly off compared to our user defined functon but not by much.

```r
################################## Mohamed Changes
x = df[,2:ncol(df)]
#x[is.na(x)] = 0
x<- as.matrix(x)
x <- as(x, "realRatingMatrix")
kable(as(x, "matrix"))
```

item_1	item_2	item_3	item_4	item_5	item_6	item_7	item_8	item_9	item_10	item_11	item_12
1	2	3	4	5	NA	1	2	1	3	1	4
5	4	3	2	1	1	2	1	2	3	1	3
1	NA	1	NA	1	2	NA	NA	1	1	2	3
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	2	2	2	3	3	3	4	4	4
1	2	3	4	5	1	2	3	4	5	1	2
NA	1	2	3	NA	1	2	3	NA	1	2	3
5	NA	4	NA	3	NA	2	NA	1	NA	1	2
5	4	3	5	4	3	5	4	3	5	4	3
1	2	1	2	1	2	NA	1	2	NA	1	2

```r
rec=Recommender(x[1:nrow(x)],method="UBCF", param=list(normalize = "Z-score",method="Cosine", nn=12))
rec2 <- predict(rec, x[1:nrow(x)])
(as(rec2, "matrix")['10',])
```

```
##   item_1   item_2   item_3   item_4   item_5   item_6   item_7   item_8
##       NA       NA       NA       NA       NA       NA 1.504270       NA
##   item_9  item_10  item_11  item_12
##       NA 1.507037       NA       NA
```

```r
############################
rec=Recommender(x[1:nrow(x)],method="UBCF", param=list(normalize = "Z-score",method="Jaccard", nn=5))
rec2 <- predict(rec, x[1:nrow(x)])
as(rec2, "matrix")['10',]
```

```
##   item_1   item_2   item_3   item_4   item_5   item_6   item_7   item_8
##       NA       NA       NA       NA       NA       NA 1.371636       NA
##   item_9  item_10  item_11  item_12
##       NA 1.877474       NA       NA
```

```r
############################
rec=Recommender(x[1:nrow(x)],method="UBCF", param=list(normalize = "Z-score",method="Jaccard"))
rec2 <- predict(rec, x[1:nrow(x)])
as(rec2, "matrix")['10',]
```

```
##   item_1   item_2   item_3   item_4   item_5   item_6   item_7   item_8
##       NA       NA       NA       NA       NA       NA 1.445948       NA
##   item_9  item_10  item_11  item_12
##       NA 1.721535       NA       NA
```

```r
##################

rec=Recommender(x[1:nrow(x)],method="IBCF", param=list(normalize = "Z-score",method="Cosine"))
rec2 <- predict(rec, x[1:nrow(x)])
as(rec2, "matrix")['10',]
```

```
##   item_1   item_2   item_3   item_4   item_5   item_6   item_7   item_8
##       NA       NA       NA       NA       NA       NA 1.214661       NA
##   item_9  item_10  item_11  item_12
##       NA 1.483787       NA       NA
```

```r
##################
```