——— DATA 643, Final Project by Burton Machado. ———-

***** Recommender System with Time and Social context. *****

Project description:

We will develop a UBCF recommender system that incorporates a Time based context in the form of days of the week. It additionally and also incorporates a Social context in the form of a relationship network. The social context is based on the connection between the users. For the purpose of this project we will consider users that are directly connected (only one hop). We will use the igraph package for analyzing the social network. In this example we will rate the movies for users using existing ratings. We will first rate a movie m2 for user u1 using the ratings data for all days of the week. Next, we will rate the movie m2 for user u1 using just the ratings for Saturdays. We will also include information about the social context in the form of how many hops users are away from each other. We will be building the whole Distance matrix for the Network, but for the sake of simplicity, we will take into account users within one hop. Here we show that users who watch movies on weekends (specifically a Saturday) are more thorough and selective in their research and hence end up liking the movies that they watch. So we end up having a watch (>3) ratings with just Saturdays data than with data for the whole week. Users who watch movies throughout the week tend to just watch movies casually and their ratings are random. We will also see that user u1 is more close to users u21 and u27, but this fact is obscured by the ratings data for other days of the week. When we use the ratings Data, for just Saturdays, and the Social context in the form of number of hops, we bring this fact out. This will be seen in the similarity matrices that will be created, one with the ratings for the whole week and one for just the ratings for Saturdays. The code is modular and it is easy to use other days of the week and also other users and movies. Altogether we have 27 users and 25 movies. The data has been artificially created. It is contained in three files, one with the ratings permovie for each user, the other with the Day of the week that a particular movie was rated by a particular user, and one with the relationship graph between the users. We could use the ideas, most of the code, in this project to build a full-fledged Recommender system. In this project, we will produce all the data for similarity and social context, but use it manually to do our prediction. The data files are available on github.

Function definitions:

```
# The following function returns the ratings for a specific day of the week denoted by the 'day' argume
# You have to pass in the Ratings data frame (df_input) and the Context data frame (df_context).
getRatingsByDay <- function(day, df_input, df_context){
  df_temp <- df_input[,2:ncol(df_input)]
  df_temp[df_context[,2:ncol(df_context)] != day] <- NA
  df_temp <- cbind(u=df_input[,1],df_temp)
  return(df_temp)
}

# The following function creates the similarity matrix from the Ratings data frame (df_r) and Mean-cent
get_similarity_matrix <- function(df_r, df_mc){
  #Construct the similarity matrix
  similarity_Matrix <- data.frame(names=df_r[1:nrow(df_r)-1,1])
  for(i in 1:nrow(df_mc)){
```

```
    temp <- c()
    for(j in 1:nrow(df_mc)){
      product <- sum(df_mc[i,1:(ncol(df_mc)-2)]*df_mc[j,1:(ncol(df_mc)-2)],na.rm = TRUE)/
        (df_mc$distance[i]*df_mc$distance[j])
      temp <- append(temp, product)
    }
    similarity_Matrix <- cbind(similarity_Matrix, temp)
  }
  names(similarity_Matrix)<-c('names', as.character(similarity_Matrix[,1]))
  return(similarity_Matrix)
}

# Helper function to bind the User means.
bind_user_means <- function(df_r){
  userMeans <- c()
  for(i in 1:nrow(df_r)){
    rowWithoutNA <- df_r[i, 2:ncol(df_r)][!is.na(df_r[i, 2:ncol(df_r)])]
    userMeans <- append(userMeans, round(mean(rowWithoutNA), 2))
  }
  df_r <- cbind(df_r, userMeans)
  return(df_r)
}

# Helper function to bind the Movie means.
bind_movie_means <- function(df_r){
  movieMeans <- c(NA)
  for(i in 2:ncol(df_r)){
    colWithoutNA <- df_r[,i][!is.na(df_r[,i])]
    movieMeans <- append(movieMeans, round(mean(colWithoutNA), 2))
  }
  df_r <- rbind(df_r, movieMeans)
  df_r[nrow(df_r),1] <- 'movieMeans'
  return(df_r)
}

# Function to generate Mean-centered data frame from a Ratings data frame (df_r).
get_mean_centered <- function(df_r){
  df_Mean_Centered <- df_r[,2:ncol(df_ratings)]-df_r$userMeans
  df_Mean_Centered <- df_Mean_Centered[,1:ncol(df_Mean_Centered)-1]
  df_Mean_Centered <- df_Mean_Centered[1:nrow(df_Mean_Centered)-1,]
  return(df_Mean_Centered)
}

# Function to canculate the distance and bind it to the data frame.
get_distance <- function(df_r, df_mc){
  distance <- c()
  for( i in 1:nrow(df_mc)){
    distance <- append(distance, sum(df_mc[i,]**2, na.rm = TRUE)**.5)
  }
  #Create mean centered data frame with the distance
  df_Mean_Centered_and_Distance <- cbind(df_mc, distance)
  df_Mean_Centered_and_Distance <- cbind(df_Mean_Centered_and_Distance, userMeans=df_r$userMeans[1:leng
  return(df_Mean_Centered_and_Distance)
```

```
}
```

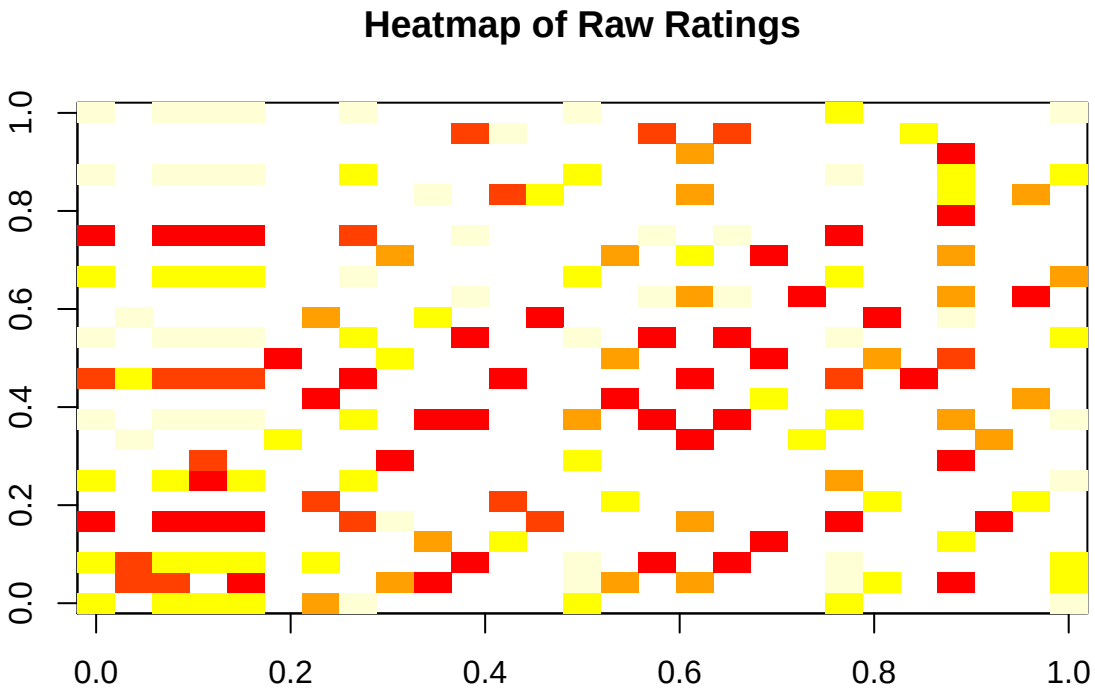
Data Load:

Our data set consists of 3 files, one for User Ratings, one for User Time Context (on which day they watched a movie) and one for User Social Context (Who is connected to whom). We have 27 Users and 25 Movies in our data set. The data set has been artificially created, with random values.

We will first load the Ratings data from the Ratings file.

```
# Please set the working directory to the location of the project files.
setwd('/Users/burton/001-Semester_05_CUNY/643_Recommender_Systems/Week_05')
# Create a data frame from the csv file.
#Get the ratings for all days of the Week
df_ratings <- read.csv('my_movies_ratings.csv', header = TRUE, stringsAsFactors = FALSE)
movie_matrix <- as.matrix(df_ratings[,2:ncol(df_ratings)])
```

Below is the heatmap of the movie ratings.



Below are the first 10 records of the Ratings data frame.

Table 1: Table continues below

u	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14
u1	4	NA	4	NA	1	NA	4	NA	NA	5	NA	2	NA	5
u2	NA	2	2	NA	NA	NA	NA	NA	5	NA	NA	4	NA	NA
u3	4	2	4	NA	1	NA	4	NA	NA	5	NA	2	NA	5
u4	4	NA	4	NA	1	NA	1	2	NA	5	NA	2	NA	5

u	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14
u5	4	1	4	NA	1	NA	4	NA	NA	5	NA	2	NA	5
u6	NA	NA	NA	NA	NA	NA	NA	NA	4	NA	NA	NA	1	NA
u7	3	NA	4	NA	NA	2	NA	NA	NA	NA	1	NA	NA	NA
u8	5	NA	NA	NA	2	NA	4	NA	NA	4	NA	1	NA	4
u9	NA	3	NA	NA	5	NA	NA	1	NA	NA	NA	NA	4	NA
u10	NA	1	NA	3	NA	NA	NA	NA	NA	1	NA	NA	NA	NA

m15	m16	m17	m18	m19	m20	m21	m22	m23	m24	m25
NA	NA	4	NA	1	NA	NA	5	NA	NA	5
5	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	4	NA	1	NA	NA	5	NA	NA	5
NA	NA	4	NA	1	NA	NA	5	NA	NA	5
NA	NA	4	NA	1	NA	NA	5	NA	NA	5
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	5	NA	2	NA	NA	4	NA	NA	5
NA	NA	NA	3	NA	NA	NA	NA	NA	NA	NA
4	NA	NA	NA	NA	NA	5	NA	NA	NA	NA

We next read in the Context information. This contains the days of the week on which movies were watched.

Below are the first 10 records of the Ratings Context data frame.

Table 3: Table continues below

u	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14
u1	Sat	NA	Sat	NA	Tue	NA	Sat	NA	NA	Sat	NA	Wed	NA	Sat
u2	NA	Tue	Sat	NA	NA	NA	NA	NA	Mon	NA	NA	Tue	NA	NA
u3	Wed	mon	Thu	NA	Sat	NA	Wed	NA	NA	Fri	NA	Sat	NA	Mon
u4	Tue	NA	Mon	NA	Sat	NA	Sat	Sat	NA	Mon	NA	Mon	NA	Fri
u5	Thu	Tue	Sun	NA	Sat	NA	Mon	NA	NA	Thu	NA	Wed	NA	Thu
u6	NA	NA	NA	NA	NA	NA	NA	NA	Tue	NA	NA	NA	Sat	NA
u7	Fri	NA	Mon	NA	NA	Sat	NA	NA	NA	NA	Sat	NA	NA	NA
u8	Sat	NA	NA	NA	Mon	NA	Sat	NA	NA	Sat	NA	Fri	NA	Sat
u9	NA	Mon	NA	NA	Tue	NA	NA	Sat	NA	NA	NA	NA	Mon	NA
u10	NA	Tue	NA	Thu	NA	NA	NA	NA	NA	Sat	NA	NA	NA	NA

m15	m16	m17	m18	m19	m20	m21	m22	m23	m24	m25
NA	NA	Sat	NA	Tue	NA	NA	Sat	NA	NA	Sat
Tue	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	Tue	NA	Mon	NA	NA	Fri	NA	NA	Sun
NA	NA	Fri	NA	Sat	NA	NA	Wed	NA	NA	Tue
NA	NA	Thu	NA	Fri	NA	NA	Thu	NA	NA	Thu
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Sat	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	Sat	NA	Thu	NA	NA	Sat	NA	NA	Sat
NA	NA	NA	Mon	NA	NA	NA	NA	NA	NA	NA
Mon	NA	NA	NA	NA	NA	Tue	NA	NA	NA	NA

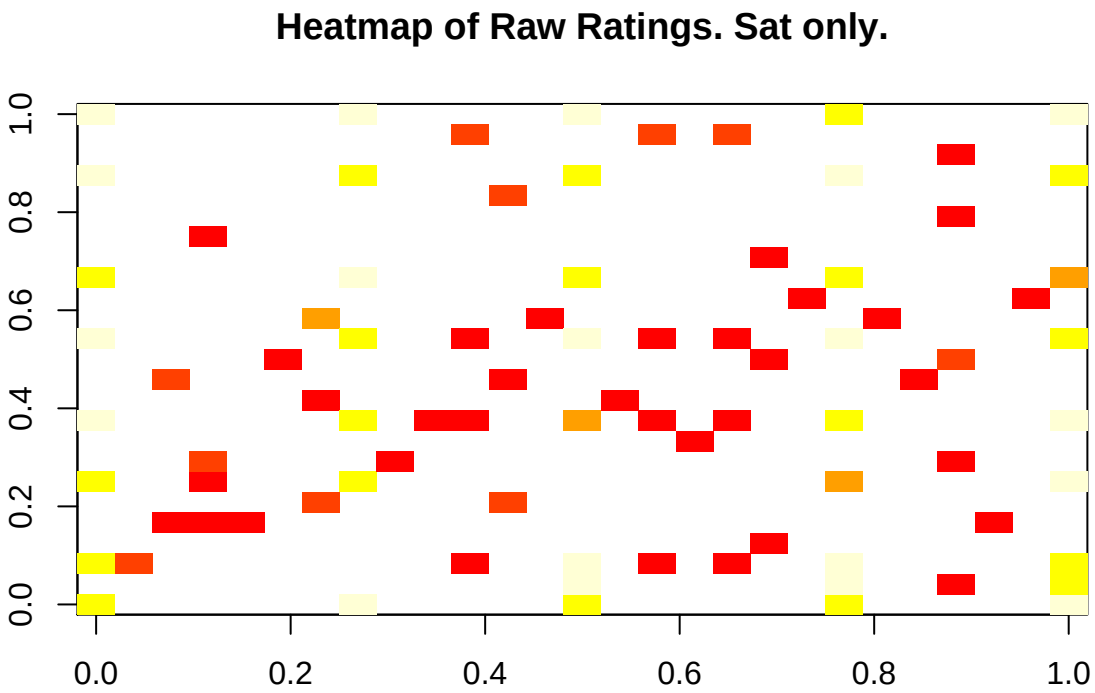
Contextual Pre-filtering (Time-based Context):

Create a data frame with just the Ratings for Saturday. This is also called a contextual pre-filtering Query, and the method is called contextual pre-filtering. This is done by joining the Ratings Matrix with the Context Matrix.

```
#Get the ratings just for Saturday.
df_ratings_sat <- getRatingsByDay('Sat', df_ratings, df_context)
```

```
movie_matrix_sat <- as.matrix(df_ratings_sat[,2:ncol(df_ratings_sat)])
```

Below is the heatmap of the Movie Ratings.



Observe that the Saturday-only Ratings are more sparse than Ratings for all days of the week. This is as expected.

Below are the first 10 records of the Saturday Ratings data frame.

Table 5: Table continues below

u	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14
u1	4	NA	4	NA	NA	NA	4	NA	NA	5	NA	NA	NA	5
u2	NA	NA	2	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u3	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	2	NA	NA
u4	NA	NA	NA	NA	1	NA	1	2	NA	NA	NA	NA	NA	NA
u5	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
u6	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	NA
u7	NA	NA	NA	NA	NA	2	NA	NA	NA	NA	1	NA	NA	NA
u8	5	NA	NA	NA	NA	NA	4	NA	NA	4	NA	NA	NA	4
u9	NA	NA	NA	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA
u10	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	NA	NA	NA	NA

m15	m16	m17	m18	m19	m20	m21	m22	m23	m24	m25
NA	NA	4	NA	NA	NA	NA	5	NA	NA	5
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	5	NA	NA	NA	NA	4	NA	NA	5
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Contextual Pre-filtering (Social Network-based Context):

We will now read in the relationship Network data using the igraph package.

```
# We then read in the relationships between users. Who is connected to whom.
relations_raw = read.csv("my_movies_social_context.csv", sep=",", row.names=1, header = TRUE)
relations <- as.matrix(relations_raw)
colnames(relations) <- paste('u',1:27, sep = "")
relationsMatrix <- relations
```

Below are the first 10 records of the Relations (Social Context) data frame.

```
pander(head(relationsMatrix, 10))
```

Table 7: Table continues below

	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12	u13
u1	1	0	0	0	0	0	0	0	0	0	0	0	0
u2	0	1	1	0	0	0	0	0	0	0	0	0	0
u3	0	0	0	1	0	0	0	0	0	0	0	0	0
u4	0	1	0	0	1	0	0	0	0	0	0	0	0
u5	0	0	0	0	0	0	1	0	0	0	0	0	0

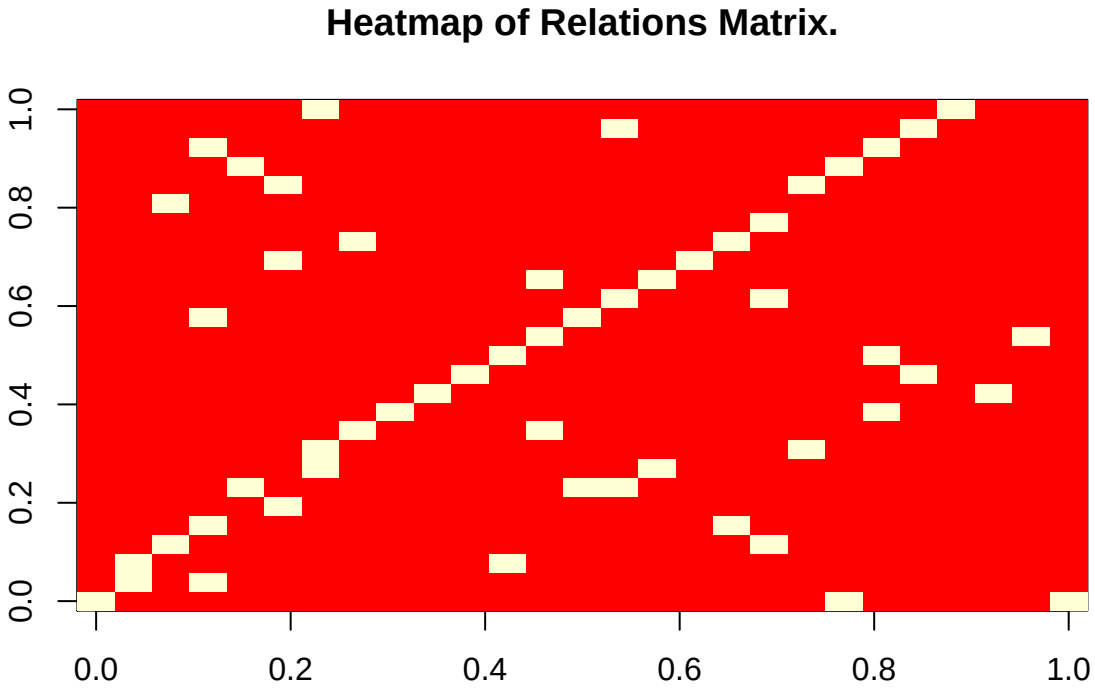
	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12	u13
u6	0	0	0	0	0	1	0	0	0	0	0	0	0
u7	0	0	0	0	0	0	0	1	1	0	0	0	0
u8	0	0	0	0	0	0	0	0	0	1	0	0	0
u9	0	0	0	0	0	0	0	0	0	0	1	0	0
u10	0	0	0	0	0	0	0	0	0	0	0	1	0

	u14	u15	u16	u17	u18	u19	u20	u21	u22	u23	u24	u25	u26	u27
u1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u3	0	0	0	0	0	0	0	0	1	0	0	0	0	0
u4	0	0	1	0	0	0	0	0	0	0	0	1	0	0
u5	0	0	0	0	0	0	0	0	0	0	1	0	0	0
u6	0	0	0	0	0	1	0	0	0	1	0	0	0	0
u7	0	0	0	0	0	0	0	0	0	0	0	0	0	1
u8	0	0	0	0	0	0	1	0	0	0	0	0	0	0
u9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u10	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note that a 1 in a user column will link that user to another user having a 1 in the same column. That is how the user network is created. For e.g. users u2 and u4 are linked through the second column of the matrix.

Below is the heatmap of the Relations (Social Context) Matrix. It gives a brief high-level view of the links between users.

```
image(relationsMatrix, main = "Heatmap of Relations Matrix.")
```

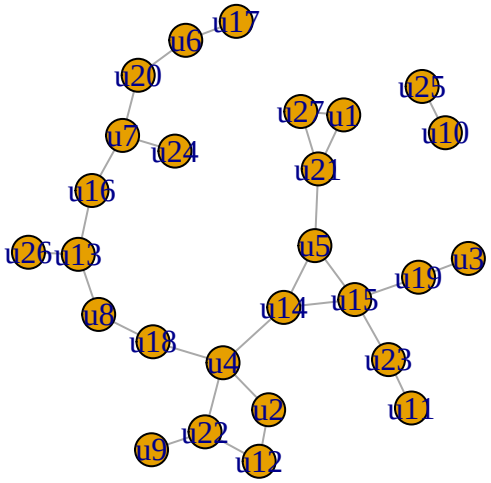


```
# Create a Boolean matrix
relationsMatrix[relationsMatrix>=1] <- 1
# Get the term-term adjacency matrix
relationsAdjacencyMatrix <- relationsMatrix %*% t(relationsMatrix)
# inspect terms numbered 5 to 10
relationsAdjacencyMatrix[5:10,5:10]
```

```
##      u5 u6 u7 u8 u9 u10
## u5    2  0  0  0  0   0
## u6    0  3  0  0  0   0
## u7    0  0  3  0  0   0
## u8    0  0  0  2  0   0
## u9    0  0  0  0  1   0
## u10   0  0  0  0  0   1
```

```
# Construct the grpah object from the above matrix
graphObject <- graph.adjacency(relationsAdjacencyMatrix, weighted=T, mode = 'undirected')
# remove loops
graphObject <- simplify(graphObject)
# set labels and degrees of vertices
V(graphObject)$label <- V(graphObject)$name
V(graphObject)$degree <- degree(graphObject)
```

We will now display the user Social Network graph.



Following is the Betweenness matrix for the Network.

Table 9: Table continues below

u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12	u13	u14	u15
0	10.5	0	181	63	23	83	128	0	0	0	1	125	135	84

u16	u17	u18	u19	u20	u21	u22	u23	u24	u25	u26	u27
95	0	135	23	44	44	33.5	23	0	0	0	0

Following is the Distance matrix for the Network. Note that the distances between u1, u21 and u27 are 1. We will later use this fact in our contextual filter query.

Table 11: Table continues below

	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12	u13
u1	0	5	5	4	2	11	9	6	6	Inf	5	6	7
u2	5	0	5	1	3	8	6	3	3	Inf	5	1	4
u3	5	5	0	4	3	11	9	6	6	Inf	4	6	7
u4	4	1	4	0	2	7	5	2	2	Inf	4	2	3
u5	2	3	3	2	0	9	7	4	4	Inf	3	4	5
u6	11	8	11	7	9	0	2	5	9	Inf	11	9	4
u7	9	6	9	5	7	2	0	3	7	Inf	9	7	2
u8	6	3	6	2	4	5	3	0	4	Inf	6	4	1
u9	6	3	6	2	4	9	7	4	0	Inf	6	2	5
u10	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf
u11	5	5	4	4	3	11	9	6	6	Inf	0	6	7
u12	6	1	6	2	4	9	7	4	2	Inf	6	0	5
u13	7	4	7	3	5	4	2	1	5	Inf	7	5	0
u14	3	2	3	1	1	8	6	3	3	Inf	3	3	4
u15	3	3	2	2	1	9	7	4	4	Inf	2	4	5
u16	8	5	8	4	6	3	1	2	6	Inf	8	6	1
u17	12	9	12	8	10	1	3	6	10	Inf	12	10	5
u18	5	2	5	1	3	6	4	1	3	Inf	5	3	2
u19	4	4	1	3	2	10	8	5	5	Inf	3	5	6
u20	10	7	10	6	8	1	1	4	8	Inf	10	8	3
u21	1	4	4	3	1	10	8	5	5	Inf	4	5	6
u22	5	2	5	1	3	8	6	3	1	Inf	5	1	4
u23	4	4	3	3	2	10	8	5	5	Inf	1	5	6
u24	10	7	10	6	8	3	1	4	8	Inf	10	8	3
u25	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1	Inf	Inf	Inf
u26	8	5	8	4	6	5	3	2	6	Inf	8	6	1
u27	1	5	5	4	2	11	9	6	6	Inf	5	6	7

	u14	u15	u16	u17	u18	u19	u20	u21	u22	u23	u24	u25	u26	u27
u1	3	3	8	12	5	4	10	1	5	4	10	Inf	8	1
u2	2	3	5	9	2	4	7	4	2	4	7	Inf	5	5
u3	3	2	8	12	5	1	10	4	5	3	10	Inf	8	5
u4	1	2	4	8	1	3	6	3	1	3	6	Inf	4	4
u5	1	1	6	10	3	2	8	1	3	2	8	Inf	6	2
u6	8	9	3	1	6	10	1	10	8	10	3	Inf	5	11
u7	6	7	1	3	4	8	1	8	6	8	1	Inf	3	9
u8	3	4	2	6	1	5	4	5	3	5	4	Inf	2	6
u9	3	4	6	10	3	5	8	5	1	5	8	Inf	6	6
u10	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1	Inf	Inf
u11	3	2	8	12	5	3	10	4	5	1	10	Inf	8	5
u12	3	4	6	10	3	5	8	5	1	5	8	Inf	6	6
u13	4	5	1	5	2	6	3	6	4	6	3	Inf	1	7
u14	0	1	5	9	2	2	7	2	2	2	7	Inf	5	3
u15	1	0	6	10	3	1	8	2	3	1	8	Inf	6	3
u16	5	6	0	4	3	7	2	7	5	7	2	Inf	2	8
u17	9	10	4	0	7	11	2	11	9	11	4	Inf	6	12

	u14	u15	u16	u17	u18	u19	u20	u21	u22	u23	u24	u25	u26	u27
u18	2	3	3	7	0	4	5	4	2	4	5	Inf	3	5
u19	2	1	7	11	4	0	9	3	4	2	9	Inf	7	4
u20	7	8	2	2	5	9	0	9	7	9	2	Inf	4	10
u21	2	2	7	11	4	3	9	0	4	3	9	Inf	7	1
u22	2	3	5	9	2	4	7	4	0	4	7	Inf	5	5
u23	2	1	7	11	4	2	9	3	4	0	9	Inf	7	4
u24	7	8	2	4	5	9	2	9	7	9	0	Inf	4	10
u25	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf
u26	5	6	2	6	3	7	4	7	5	7	4	Inf	0	8
u27	3	3	8	12	5	4	10	1	5	4	10	Inf	8	0

Note that the distances (Hops) between u1, u21 and u27 are 1. We will be using this fact to give more weightage to u21 and u27 with respect to u1. One way this can be done is to increase the similarity between u1 and u21, and u1 and u27.

Build the Recommedations:

```
# Here we will calculate the user means and bind them to the data frame
df_ratings <- bind_user_means(df_ratings)
df_ratings_sat <- bind_user_means(df_ratings_sat)

# Here we will calculate the movie means and bind them to the data frame
df_ratings <- bind_movie_means(df_ratings)
df_ratings_sat <- bind_movie_means(df_ratings_sat)
```

```
## Warning in `[<-.factor`(`*tmp*`, iseq, value = "movieMeans"): invalid
## factor level, NA generated
```

Following is the Ratings data set (first 10 rows) with the Movie means.

Table 13: Table continues below

u	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14
u1	4	NA	4	NA	1	NA	4	NA	NA	5	NA	2	NA	5
u2	NA	2	2	NA	NA	NA	NA	NA	5	NA	NA	4	NA	NA
u3	4	2	4	NA	1	NA	4	NA	NA	5	NA	2	NA	5
u4	4	NA	4	NA	1	NA	1	2	NA	5	NA	2	NA	5
u5	4	1	4	NA	1	NA	4	NA	NA	5	NA	2	NA	5
u6	NA	NA	NA	NA	NA	NA	NA	NA	4	NA	NA	NA	1	NA
u7	3	NA	4	NA	NA	2	NA	NA	NA	NA	1	NA	NA	NA
u8	5	NA	NA	NA	2	NA	4	NA	NA	4	NA	1	NA	4
u9	NA	3	NA	NA	5	NA	NA	1	NA	NA	NA	NA	4	NA
u10	NA	1	NA	3	NA	NA	NA	NA	NA	1	NA	NA	NA	NA

m15	m16	m17	m18	m19	m20	m21	m22	m23	m24	m25	userMeans
NA	NA	4	NA	1	NA	NA	5	NA	NA	5	3.64
5	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	3.6
NA	NA	4	NA	1	NA	NA	5	NA	NA	5	3.5
NA	NA	4	NA	1	NA	NA	5	NA	NA	5	3.25
NA	NA	4	NA	1	NA	NA	5	NA	NA	5	3.42
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.5
3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.6
NA	NA	5	NA	2	NA	NA	4	NA	NA	5	3.6
NA	NA	NA	3	NA	NA	NA	NA	NA	NA	NA	3.2
4	NA	NA	NA	NA	NA	5	NA	NA	NA	NA	2.8

Following is the Saturdays Ratings data set (first 10 rows) with the Movie means.

Table 15: Table continues below

u	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14
u1	4	NA	4	NA	NA	NA	4	NA	NA	5	NA	NA	NA	5
u2	NA	NA	2	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u3	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	2	NA	NA
u4	NA	NA	NA	NA	1	NA	1	2	NA	NA	NA	NA	NA	NA
u5	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
u6	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	NA
u7	NA	NA	NA	NA	NA	2	NA	NA	NA	NA	1	NA	NA	NA
u8	5	NA	NA	NA	NA	NA	4	NA	NA	4	NA	NA	NA	4
u9	NA	NA	NA	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA
u10	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	NA	NA	NA	NA

m15	m16	m17	m18	m19	m20	m21	m22	m23	m24	m25	userMeans
NA	NA	4	NA	NA	NA	NA	5	NA	NA	5	4.5
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.5
NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	1.25
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1
3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2
NA	NA	5	NA	NA	NA	NA	4	NA	NA	5	4.43
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1

```
# We now calculate the mean-centered values and create another dataframe.
df_Mean_Centered <- get_mean_centered(df_ratings)
df_Mean_Centered_sat <- get_mean_centered(df_ratings_sat)

# Print the mean-centered dataframe in a table.
#kable(df_Mean_Centered)
```

```
# Calculate the distances
df_Mean_Centered_and_Distance <- get_distance(df_ratings, df_Mean_Centered)
df_Mean_Centered_and_Distance_sat <- get_distance(df_ratings_sat, df_Mean_Centered_sat)
```

Following is the Ratings data set (first 10 rows) that is Mean-centered and with the Distances.

Table 17: Table continues below

m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14
0.36	NA	0.36	NA	-2.64	NA	0.36	NA	NA	1.36	NA	-1.64	NA	1.36
NA	-1.6	-1.6	NA	NA	NA	NA	NA	1.4	NA	NA	0.4	NA	NA
0.5	-1.5	0.5	NA	-2.5	NA	0.5	NA	NA	1.5	NA	-1.5	NA	1.5
0.75	NA	0.75	NA	-2.25	NA	-2.25	-1.25	NA	1.75	NA	-1.25	NA	1.75
0.58	-2.42	0.58	NA	-2.42	NA	0.58	NA	NA	1.58	NA	-1.42	NA	1.58
NA	NA	NA	NA	NA	NA	NA	NA	1.5	NA	NA	NA	-1.5	NA
0.4	NA	1.4	NA	NA	-0.6	NA	NA	NA	NA	-1.6	NA	NA	NA
1.4	NA	NA	NA	-1.6	NA	0.4	NA	NA	0.4	NA	-2.6	NA	0.4
NA	-0.2	NA	NA	1.8	NA	NA	-2.2	NA	NA	NA	NA	0.8	NA
NA	-1.8	NA	0.2	NA	NA	NA	NA	NA	-1.8	NA	NA	NA	NA

m15	m16	m17	m18	m19	m20	m21	m22	m23	m24	m25	distance	userMeans
NA	NA	0.36	NA	-2.64	NA	NA	1.36	NA	NA	1.36	4.954	3.64
1.4	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	3.033	3.6
NA	NA	0.5	NA	-2.5	NA	NA	1.5	NA	NA	1.5	5.196	3.5
NA	NA	0.75	NA	-2.25	NA	NA	1.75	NA	NA	1.75	5.679	3.25
NA	NA	0.58	NA	-2.42	NA	NA	1.58	NA	NA	1.58	5.56	3.42
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.121	2.5
0.4	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.28	2.6
NA	NA	1.4	NA	-1.6	NA	NA	0.4	NA	NA	1.4	4.29	3.6
NA	NA	NA	-0.2	NA	NA	NA	NA	NA	NA	NA	2.966	3.2
1.2	NA	NA	NA	NA	NA	2.2	NA	NA	NA	NA	3.578	2.8

Following is the Saturdays Ratings data set (first 10 rows) that is Mean-centered and with the Distances.

Table 19: Table continues below

m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14
-0.5	NA	-0.5	NA	NA	NA	-0.5	NA	NA	0.5	NA	NA	NA	0.5
NA	NA	0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	-0.5	NA	NA	NA	NA	NA	NA	0.5	NA	NA
NA	NA	NA	NA	-0.25	NA	-0.25	0.75	NA	NA	NA	NA	NA	NA

m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14
NA	NA	NA	NA	0	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	NA
NA	NA	NA	NA	NA	0	NA	NA	NA	NA	-1	NA	NA	NA
0.57	NA	NA	NA	NA	NA	-0.43	NA	NA	-0.43	NA	NA	NA	-0.43
NA	NA	NA	NA	NA	NA	NA	0	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	0	NA	NA	NA	NA

m15	m16	m17	m18	m19	m20	m21	m22	m23	m24	m25	distance	userMeans
NA	NA	-0.5	NA	NA	NA	NA	0.5	NA	NA	0.5	1.414	4.5
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	2
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0.7071	1.5
NA	NA	NA	NA	-0.25	NA	NA	NA	NA	NA	NA	0.866	1.25
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	1
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	1
1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.414	2
NA	NA	0.57	NA	NA	NA	NA	-0.43	NA	NA	0.57	1.309	4.43
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	1
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	1

```
#Construct the similarity matrix
similarity_Matrix_whole <- get_similarity_matrix(df_ratings, df_Mean_Centered_and_Distance)
similarity_Matrix_sat <- get_similarity_matrix(df_ratings_sat, df_Mean_Centered_and_Distance_sat)
```

Following is the Similarity Matrix for the whole data set.

Table 21: Table continues below

names	u1	u2	u3	u4	u5	u6	u7	u8
u1	1	-0.08198	0.9532	0.8335	0.8907	0	0.05736	0.8188
u2	-0.08198	1	0.06345	-0.09869	0.1409	0.3264	-0.2429	-0.07993
u3	0.9532	0.06345	1	0.8006	0.9864	0	0.07596	0.7807
u4	0.8335	-0.09869	0.8006	1	0.7514	0	0.1042	0.665
u5	0.8907	0.1409	0.9864	0.7514	1	0	0.08234	0.7295
u6	0	0.3264	0	0	0	1	0	0
u7	0.05736	-0.2429	0.07596	0.1042	0.08234	0	1	0.05725
u8	0.8188	-0.07993	0.7807	0.665	0.7295	0	0.05725	1
u9	-0.3233	0.03556	-0.2725	-0.07717	-0.2347	-0.1907	0	-0.2263
u10	-0.1381	0.4202	0	-0.155	0.07601	0	0.05883	-0.04692
u11	-0.5128	0.1792	-0.5012	-0.4785	-0.4749	0	-0.2085	-0.2745
u12	0.1813	-0.07223	0.1581	0.1206	0.1399	0	0.06405	0.332
u13	0.0814	-0.2842	0.0735	0.06052	0.06649	0	-0.108	0.05698
u14	-0.04356	-0.3534	-0.1424	-0.01233	-0.1907	0	0.1767	-0.04453
u15	0	-0.04815	-0.02635	0	-0.03973	-0.06455	0.4323	0
u16	-0.5128	0.1792	-0.5012	-0.4785	-0.4749	0	-0.2085	-0.2745
u17	0.1189	-0.414	0.06241	0.07342	0.02956	-0.4082	0	0.2994

names	u1	u2	u3	u4	u5	u6	u7	u8
u18	-0.5128	0.1792	-0.5012	-0.4785	-0.4749	0	-0.2085	-0.2745
u19	0	0	0	0	0	0.2041	-0.6076	0
u20	0	0.3264	0	0	0	0.5	0	0
u21	0.8824	-0.3419	0.7518	0.8122	0.6515	0	0.1894	0.7099
u22	0	-0.5922	-0.1179	0	-0.1777	0	-0.2506	0
u23	0.2341	-0.09325	0.2041	0.1556	0.1806	0	0	0.4286
u24	0.09571	0.4109	0.2019	0.1847	0.2517	0.09951	0.08585	0.03251
u25	0.3768	0.3264	0.3402	0.2802	0.3078	0.5	0	0.2638
u26	0	0	0	0	0	0	-0.2314	0
u27	0.0817	0.1741	0.1304	-0.09333	0.1517	0	-0.04254	0.03334

Table 22: Table continues below

u9	u10	u11	u12	u13	u14	u15	u16
-0.3233	-0.1381	-0.5128	0.1813	0.0814	-0.04356	0	-0.5128
0.03556	0.4202	0.1792	-0.07223	-0.2842	-0.3534	-0.04815	0.1792
-0.2725	0	-0.5012	0.1581	0.0735	-0.1424	-0.02635	-0.5012
-0.07717	-0.155	-0.4785	0.1206	0.06052	-0.01233	0	-0.4785
-0.2347	0.07601	-0.4749	0.1399	0.06649	-0.1907	-0.03973	-0.4749
-0.1907	0	0	0	0	0	-0.06455	0
0	0.05883	-0.2085	0.06405	-0.108	0.1767	0.4323	-0.2085
-0.2263	-0.04692	-0.2745	0.332	0.05698	-0.04453	0	-0.2745
1	0.03392	0	0	-0.09269	0.09978	0.01231	0
0.03392	1	0.1709	-0.1293	0.2689	0.166	-0.04593	0.1709
0	0.1709	1	-0.0758	0	-0.001698	0	1
0	-0.1293	-0.0758	1	-0.1882	0	-0.1333	-0.0758
-0.09269	0.2689	0	-0.1882	1	0	0	0
0.09978	0.166	-0.001698	0	0	1	0.03058	-0.001698
0.01231	-0.04593	0	-0.1333	0	0.03058	1	0
0	0.1709	1	-0.0758	0	-0.001698	0	1
0.03988	0.01514	0.07666	0.2832	0.08399	0.04536	0.05725	0.07666
0	0.1709	1	-0.0758	0	-0.001698	0	1
-0.05839	-0.01614	0	-0.1054	0	0	-0.7642	0
0	0	-0.4003	0	0	0	0	-0.4003
-0.3328	-0.1855	-0.5145	0.1734	0.07896	0.1834	0.02597	-0.5145
-0.02752	-0.4793	0	-0.09938	0.5027	0.1368	0.2609	0
0	0	-0.08006	0.8607	0	0	0	-0.08006
0.2412	0.4932	0.0157	0.034	-0.08535	-0.1518	-0.03854	0.0157
-0.4291	0	0	0	0.108	0	0	0
0	0.07054	-0.4546	-0.1675	0.08868	0	0.2199	-0.4546
0.01112	-0.08553	-0.001698	0	0	-0.1947	-0.01506	-0.001698

Table 23: Table continues below

u17	u18	u19	u20	u21	u22	u23	u24	u25
0.1189	-0.5128	0	0	0.8824	0	0.2341	0.09571	0.3768
-0.414	0.1792	0	0.3264	-0.3419	-0.5922	-0.09325	0.4109	0.3264
0.06241	-0.5012	0	0	0.7518	-0.1179	0.2041	0.2019	0.3402
0.07342	-0.4785	0	0	0.8122	0	0.1556	0.1847	0.2802

u17	u18	u19	u20	u21	u22	u23	u24	u25
0.02956	-0.4749	0	0	0.6515	-0.1777	0.1806	0.2517	0.3078
-0.4082	0	0.2041	0.5	0	0	0	0.09951	0.5
0	-0.2085	-0.6076	0	0.1894	-0.2506	0	0.08585	0
0.2994	-0.2745	0	0	0.7099	0	0.4286	0.03251	0.2638
0.03988	0	-0.05839	0	-0.3328	-0.02752	0	0.2412	-0.4291
0.01514	0.1709	-0.01614	0	-0.1855	-0.4793	0	0.4932	0
0.07666	1	0	-0.4003	-0.5145	0	-0.08006	0.0157	0
0.2832	-0.0758	-0.1054	0	0.1734	-0.09938	0.8607	0.034	0
0.08399	0	0	0	0.07896	0.5027	0	-0.08535	0.108
0.04536	-0.001698	0	0	0.1834	0.1368	0	-0.1518	0
0.05725	0	-0.7642	0	0.02597	0.2609	0	-0.03854	0
0.07666	1	0	-0.4003	-0.5145	0	-0.08006	0.0157	0
1	0.07666	-0.142	-0.5039	0.1513	0.05528	0.4082	-0.0137	-0.5039
0.07666	1	0	-0.4003	-0.5145	0	-0.08006	0.0157	0
-0.142	0	1	0	0	0	0	-0.06003	0
-0.5039	-0.4003	0	1	0	0	0	-0.04901	0.5
0.1513	-0.5145	0	0	1	0.1161	0.2238	-0.01448	0.3655
0.05528	0	0	0	0.1161	1	0	-0.5428	0
0.4082	-0.08006	0	0	0.2238	0	1	0	0
-0.0137	0.0157	-0.06003	-0.04901	-0.01448	-0.5428	0	1	0
-0.5039	0	0	0.5	0.3655	0	0	0	1
-0.0932	-0.4546	0.09934	0.5678	0	0.2341	0	-0.02361	0
-0.02234	-0.001698	0	0	-0.1981	-0.06736	0	0.035	0

u26	u27
0	0.0817
0	0.1741
0	0.1304
0	-0.09333
0	0.1517
0	0
-0.2314	-0.04254
0	0.03334
0	0.01112
0.07054	-0.08553
-0.4546	-0.001698

u26	u27
-0.1675	0
0.08868	0
0	-0.1947
0.2199	-0.01506
-0.4546	-0.001698
-0.0932	-0.02234
-0.4546	-0.001698
0.09934	0
0.5678	0
0	-0.1981
0.2341	-0.06736
0	0
-0.02361	0.035
0	0
1	0
0	1

Following is the Similarity Matrix for the data set filtered by Saturdays.

Table 25: Table continues below

names	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10
u1	1	NA	0	0.1021	NA	NA	0	-0.3861	NA	NA
u2	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u3	0	NA	1	0.2041	NA	NA	0	0	NA	NA
u4	0.1021	NA	0.2041	1	NA	NA	0	0.09481	NA	NA
u5	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u6	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u7	0	NA	0	0	NA	NA	1	0	NA	NA
u8	-0.3861	NA	0	0.09481	NA	NA	0	1	NA	NA
u9	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u10	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u11	-0.1021	NA	0	0	NA	NA	0	0.1896	NA	NA
u12	0	NA	-0.5802	0	NA	NA	0	0	NA	NA
u13	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u14	-0.06825	NA	0	0	NA	NA	0	0.1592	NA	NA
u15	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u16	-0.1021	NA	0	0	NA	NA	0	0.1896	NA	NA
u17	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u18	-0.1021	NA	0	0	NA	NA	0	0.1896	NA	NA
u19	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u20	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u21	0.3535	NA	0	0.192	NA	NA	0	-0.1629	NA	NA
u22	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u23	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u24	0	NA	0	-0.1936	NA	NA	0	0	NA	NA
u25	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u26	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
u27	0.1768	NA	0	-0.0967	NA	NA	0	-0.1095	NA	NA

Table 26: Table continues below

u11	u12	u13	u14	u15	u16	u17	u18	u19	u20
-0.1021	0	NA	-0.06825	NA	-0.1021	NA	-0.1021	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0	-0.5802	NA	0	NA	0	NA	0	NA	NA
0	0	NA	0	NA	0	NA	0	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0	0	NA	0	NA	0	NA	0	NA	NA
0.1896	0	NA	0.1592	NA	0.1896	NA	0.1896	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1	0	NA	0.02053	NA	1	NA	1	NA	NA
0	1	NA	0	NA	0	NA	0	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0.02053	0	NA	1	NA	0.02053	NA	0.02053	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1	0	NA	0.02053	NA	1	NA	1	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1	0	NA	0.02053	NA	1	NA	1	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
-0.1458	0	NA	0.3793	NA	-0.1458	NA	-0.1458	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0	0	NA	-0.07043	NA	0	NA	0	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
-0.001443	0	NA	-0.1897	NA	-0.001443	NA	-0.001443	NA	NA

u21	u22	u23	u24	u25	u26	u27
0.3535	NA	NA	0	NA	NA	0.1768
NA	NA	NA	NA	NA	NA	NA
0	NA	NA	0	NA	NA	0
0.192	NA	NA	-0.1936	NA	NA	-0.0967
NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA
0	NA	NA	0	NA	NA	0
-0.1629	NA	NA	0	NA	NA	-0.1095
NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA
-0.1458	NA	NA	0	NA	NA	-0.001443
0	NA	NA	0	NA	NA	0
NA	NA	NA	NA	NA	NA	NA
0.3793	NA	NA	-0.07043	NA	NA	-0.1897
NA	NA	NA	NA	NA	NA	NA
-0.1458	NA	NA	0	NA	NA	-0.001443
NA	NA	NA	NA	NA	NA	NA
-0.1458	NA	NA	0	NA	NA	-0.001443
NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA

u21	u22	u23	u24	u25	u26	u27
1	NA	NA	-0.07491	NA	NA	-0.5
NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA
-0.07491	NA	NA	1	NA	NA	0.03689
NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA
-0.5	NA	NA	0.03689	NA	NA	1

```
df_Mean_Centered_and_Distance <- cbind(Reviewer=df_ratings[1:(nrow(df_ratings)-1),1],df_Mean_Centered_a
similarity_u1 <- sort(similarity_Matrix_whole$u1,decreasing = TRUE)
similarity_u1_sat <- sort(similarity_Matrix_sat$u1,decreasing = TRUE)
```

We will rate movie m2 for user u1 using ALL ratings (ratings for all days of the week). The two most similar for user u1 are u3 (0.9532) and u5(0.8907). The weights for u3 is -1.50 and for u5 is -2.42. The mean rating of u1 for m2 is 2.

```
rating_u1_m2 <- ((0.9532*(-1.50) + 0.8907*(-2.42))/(0.9532+0.8907)) + 3.64
```

We will rate movie m2 for user u1 just using the Saturday ratings. We will also use the fact the distances (Hops) between u1, u21 and u27 in the user Social Network are 1. two most similar for user u1 (for Saturday) are u21 (Similarity : 0.3535) and u27(Similarity : 0.1768). The weights for u21 is 0.67 and for u27 is -0.33. The mean rating of for m2 is 1.5.

```
rating_u1_m2_sat <- ((0.3535*(0.67) + 0.1768*(-0.33))/(0.3535+0.1768)) + 4.50
```

Following is the rating for user u1 on movie m2 taking all days of the week into context.

1.696

Following is the rating for user u1 on movie m2 taking only Saturdays ratings data and User Network into context. For the Social Network context, we use the fact the distances (Hops) between u1, u21 and u27 in the user Network are 1.

4.837

We can observe that the rating for user u1 for movie m2 is 1.7 if the whole weeks ratings data is used.

We can observe that the rating for user u1 for movie m2 is 4.84 if only Saturdays ratings data and User Network is used.

Also users u3 and u5 obscure the proximity of users u21 and u27 who are actually more close to user u1 in terms of weekend ratings similarity. This fact is brought about by the higher rating that u1 gets when we just use Saturdays ratings data. Users u21 and u27 in our data are actually real weekend users if we check the raw ratings data. So the ratings prediction changes from a low to high when we switch to Saturdays data and User Network.