

```
#####
```

```
### This program is written in R. It is fully executable as written in a R environment
```

```
#####
```

```
## Require packages and read in the the data collateral provided.
```

```
### Prediction
```

```
require(plyr)
require(reshape2)
setwd("/Users/sarpotd/Desktop/Coursera/Recommender Systems/week6/Assignment2/")
ratings <- read.csv("data/ratings.csv", header=FALSE)
colnames(ratings) <- c("user_id", "movie_id", "ratings")
users <- read.csv("data/users.csv", header=FALSE)
colnames(users) <- c("user_id", "user_name")
movies <- read.csv("data/movie-titles.csv", header=FALSE)
colnames(movies) <- c("movie_id", "movie_name")
```

```
## Data munging to change it to a form useful for manipulation
## user_ratings as ratings of users for movies in a matrix form.
```

```
temp_1 <- melt(ratings, id=c("user_id","movie_id"), measure="ratings")
temp_2 <- acast(temp_1, user_id ~ movie_id ~ variable)
user_ratings <- as.matrix(temp_2[,1])
```

```
## Backup copy of the user_ratings matrix before we do data munging on it.
```

```
user_ratings_rownames <- rownames(user_ratings)
user_ratings_bak <- user_ratings
user_ratings_bak[is.na(user_ratings)] = 0
```

```
## Calculate mean for all users. Omit any "NA"
```

```
mean_calc <- function(c) (mean(na.omit(c)))
user_ratings_mean <- as.matrix(apply(user_ratings,1,mean_calc))
rownames(user_ratings_mean) <- user_ratings_rownames
```

```
## Subtract means for all users from their individual ratings for each item.
## This gets us mean centered ratings.
```

```
sub_mean <- function(c) (c - user_ratings_mean)
temp_1 <- apply(user_ratings_bak,2,sub_mean)
rownames(temp_1) <- user_ratings_rownames
user_ratings_mean_centered <- temp_1
user_ratings_mean_centered[is.na(user_ratings)] = 0
```

```
## Calculate Euclidean Norm for each user using their mean centered ratings.
```

```

norm_calc <- function(c) (sqrt(sum(c^2)))
item_ratings_norm <- as.matrix(apply(user_ratings_mean_centered,2,norm_calc))

## Create a list fo the users and movies assigned to us

input_user_list <-
c(rep("1408",times=5),rep("2401",times=5),rep("1714",times=5),rep("4781",times=5),rep("445",times=5))
input_item_list <-
c("1891","63","752","786","8358","2024","9802","2502","424","601","9802","146","462","9741",
"8358","243","194","141","153","7443","664","1637","9802","153","788")

### This is the meat of the program. A function called final_fun is written
### This is a function to calcualte the cosine similarity with each input_user, find the
top 30 "similar" neighbors that have rated the input_item
### Then find the prediction based on the formula given the instructions of the
assignment.

final_fun <- function(c) {
input_user <- input_user_list[c]
input_item <- input_item_list[c]
cosine_fun <- function(c) (sum(c*user_ratings_mean_centered[,input_item]))
temp <- as.matrix(apply(user_ratings_mean_centered,2,cosine_fun))
norm_fun <- function(c) (temp[c]/(item_ratings_norm[input_item,]*item_ratings_norm[c]))
cos_similarity <- data.frame(mapply(norm_fun,1:nrow(temp)))
rownames(cos_similarity) <- rownames(temp)
cos_similarity[,2] <- rownames(cos_similarity)
cos_similarity[cos_similarity[,1] < 0,1] = 0

colnames(cos_similarity) <- c("cosine","item")

cos_similarity <- cos_similarity[!is.na(user_ratings[input_user,]),]
neighbors <- cos_similarity[cos_similarity$item!=input_item,]
neighbors <- neighbors[order(neighbors$cosine,decreasing=TRUE),][1:20,]

print(input_user)
print(input_item)
print(sum(user_ratings[input_user,neighbors[, "item"]] * neighbors[, "cosine"])/
sum(neighbors[, "cosine"]))
print(movies[movies$movie_id==input_item,"movie_name"],max.levels=0)
}

### Call the final_fun function on each of the user:movie combination iteratively.

mapply(final_fun,1:length(input_user_list))

```