

Ex6-Filtering

Naga Soundari Balamurugan

November 15, 2018

```
ratingData <- read.csv("rating_new.csv")
```

1. Use a Collaborative Filter approach to determine which of the 3 new games (Lego Movie, Gran Turismo 6 or TitanFall) should be recommended to Toby.

```
#Converting all the type to numeric to calculate correlation
ratingData$Super.Mario.3D.World <- as.numeric(ratingData$Super.Mario.3D.World)
ratingData$The.Last.of.US <- as.numeric(ratingData$The.Last.of.US)
ratingData$Tomb.Raider..2013. <- as.numeric(ratingData$Tomb.Raider..2013.)
ratingData$GTA.5...San.Andreas <- as.numeric(ratingData$GTA.5...San.Andreas)
ratingData$NBA.2K14 <- as.numeric(ratingData$NBA.2K14)
ratingData$Gran.Turismo.6 <- as.numeric(ratingData$Gran.Turismo.6)
ratingData$TitanFall <- as.numeric(ratingData$TitanFall)
ratingData$Lego.Movie.Game <- as.numeric(ratingData$Lego.Movie.Game)

#Transpose the data frame to find correlation between users
ratingData_Tr <- data.frame(t(ratingData[-1]))
colnames(ratingData_Tr) <- ratingData[, 1]

#Find correlation
corTest <- cor(ratingData_Tr[,1:11], use = "pairwise.complete.obs")

#Create a dataframe to store the correlation values
filtrationDF <- NULL
filtrationDF$Name <- colnames(ratingData_Tr)
filtrationDF$CorrVal <- corTest[1:11]
filtrationDF$Average <- colMeans(x=ratingData_Tr, na.rm = TRUE)
filtrationDF <- as.data.frame(filtrationDF)

#Average vote of Toby
Toby_Avg_Vote <- filtrationDF$Average[1]

##### Gran Turismo 6 #####
#Select only the rows that has rated for Gran Turismo
filtrationDF_Gran <- filtrationDF[-c(1, 5, 10), ]

#Calculate the scaling factor
scalingFactor_Gran <- sum(abs(filtrationDF_Gran$CorrVal))

#Add a column with corresponding weights
filtrationDF_Gran$weights <- (filtrationDF_Gran$CorrVal)/scalingFactor_Gran
filtrationDF_Gran <- as.data.frame(filtrationDF_Gran)
```

```

#Add a column with corresponding rating
omitted <- na.omit(ratingData$Gran.Turismo.6)
filtrationDF_Gran$Rating <- omitted

#Find the difference between rating for Gran and average vote for each user
filtrationDF_Gran$Difference <- filtrationDF_Gran$Rating - (filtrationDF_Gran$Average)

#Find weighted difference for each user
filtrationDF_Gran$WeighDiff <- filtrationDF_Gran$weights * filtrationDF_Gran$Difference

#Prediction of rating
Gran_weighedDiff <- sum(filtrationDF_Gran$WeighDiff)
PredictedVote_Gran <- Toby_Avg_Vote + Gran_weighedDiff

##### TitanFall #####
#Select only the rows that has rated for TitanFall
filtrationDF_TitanFall <- filtrationDF[-c(1, 6, 7, 8), ]

#Calculate the scaling factor
scalingFactor_TitanFall <- sum(abs(filtrationDF_TitanFall$CorrVal))

#Add a column with corresponding weights
filtrationDF_TitanFall$weights <- (filtrationDF_TitanFall$CorrVal)/scalingFactor_TitanFall
filtrationDF_TitanFall <- as.data.frame(filtrationDF_TitanFall)

#Add a column with corresponding rating
omitted <- na.omit(ratingData$TitanFall)
filtrationDF_TitanFall$Rating <- omitted

#Find the difference between rating for TitanFall and average vote for each user
filtrationDF_TitanFall$Difference <- filtrationDF_TitanFall$Rating - (filtrationDF_TitanFall$Average)

#Find weighted difference for each user
filtrationDF_TitanFall$WeighDiff <- filtrationDF_TitanFall$weights *
  filtrationDF_TitanFall$Difference

#Prediction of rating
TitanFall_weighedDiff <- sum(filtrationDF_TitanFall$WeighDiff)
PredictedVote_TitanFall <- Toby_Avg_Vote + TitanFall_weighedDiff

##### Lego Movie Game #####
#Select only the rows that has rated for Lego Movie Game
filtrationDF_Lego <- filtrationDF[-c(1, 3, 4, 11), ]

#Calculate the scaling factor
scalingFactor_Lego <- sum(abs(filtrationDF_Lego$CorrVal))

#Add a column with corresponding weights
filtrationDF_Lego$weights <- (filtrationDF_Lego$CorrVal)/scalingFactor_Lego
filtrationDF_Lego <- as.data.frame(filtrationDF_Lego)

```

```

#Add a column with corresponding rating
omitted <- na.omit(ratingData$Lego.Movie.Game)
filtrationDF_Lego$Rating <- omitted

#Find the difference between rating for Lego Movie Game and average vote for each user
filtrationDF_Lego$Difference <- filtrationDF_Lego$Rating - (filtrationDF_Lego$Average)

#Find weighted difference for each user
filtrationDF_Lego$WeighDiff <- filtrationDF_Lego$weights *
  filtrationDF_Lego$Difference

#Prediction of rating
Lego_weighedDiff <- sum(filtrationDF_Lego$WeighDiff)
PredictedVote_Lego <- Toby_Avg_Vote + Lego_weighedDiff

```

The predicted rating of the three new games for Toby are as follows: Gran Turismo 6: 4.376301
 Titanfall: 2.412996 Lego Movie Game: 0.5915214 Hence Toby could be recommended with **Gran
 Turismo 6**