Movie Recommendation System

Anuj Tanwar

08/14/2021

Recommendation System - Introduction

Recommendation Systems play a crucial and important role in this "era of abundance". For any product, there are thousands of options.Recommendation systems help to personalize a platform and help the user find something they might like. Online streaming services such as Netflix, Disney, YouTube, Amazon, etc use these recommendation systems to make the right recommendations about the movies and shows can have drastic impact on the customer base and the overall business. If customers are satisfied with the recommendations, they will spend more time using the service and would not look for other streaming services options which leads to customer retention and growth in sales and profit. Various sources say that as much as 35–40% of tech giants' revenue comes from recommendations alone.

The Netflix offered a Prize of 1 million dollar and hosted an open competition for the best recommendation algorithm to predict user ratings for films, based on previous ratings without any other information about the users or films, i.e. without the users or the films being identified except by numbers assigned for the contest.

Below is an image from researchgate.net that demonstrates a recommendation system at high level.

Problem Statement

1. Providing useful and related content out of collection of relevant and irrelevant items to users.
2. Purpose of Online Movie Recommendation System is to recommend movies based on previous ratings.
3. Given a set of users with their previous ratings for a set of movies, can we predict the rating they will assign to a movie they have not previously rated?
4. Example: Which movie will you like, If you have seen Iron Man, captain America, Hulk and Thor. Users who saw these movies also liked "Avengers.

How you addressed this problem statement

I am using Netfilx data and MovieLense data to demonstrate my approach to address the problem.

Below are the steps involved with Netflix data:

1. Download the datasets
2. Read the dataset files in different dataframes
3. Clean the data
4. Transform the data to have customer and movies related data along with Ratings given to the movies.
5. Now we can use this dataframe to do the regression and predictions

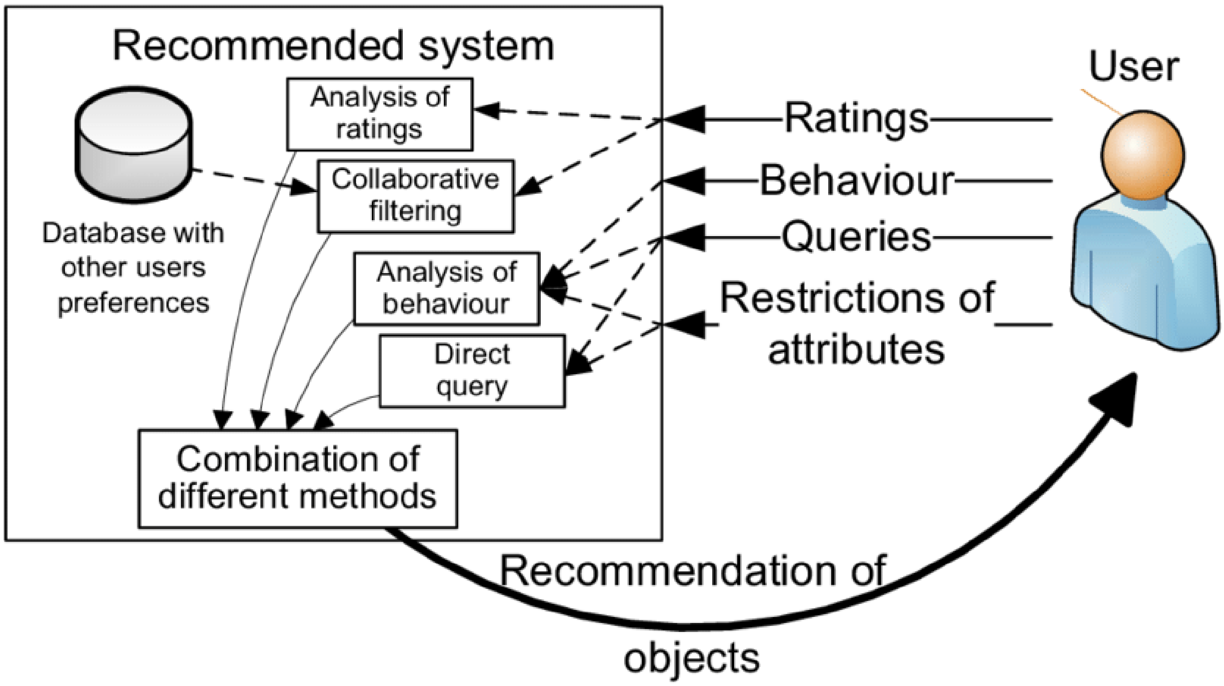

Figure 1: Recommendation System

Below are the steps involved with MovieLense data:

1. Download the datasets
2. Read the dataset files in different dataframes
3. Clean the data
4. Transform and join different dataframes from different files
5. Now we can use this dataframe to do the regression and predictions

Analysis

Netflix Data Analysis

```
netflix_titles <- read.csv("C:/Users/anujt/Documents/R/GIT/dsc520/data/netflix/movie_titles.csv",header=
names(netflix_titles) <- c('MovieID','YearOfRelease','Title')

netflix_df=read.csv("C:/Users/anujt/Documents/R/GIT/dsc520/data/netflix/combined_data_1.txt",header=FALS
netflix_df <- transform(netflix_df, MovieID=ifelse(grepl(":",CustomerID),str_replace(CustomerID,":",""),
netflix_df <- netflix_df %>% fill(MovieID)
netflix_df <- netflix_df[netflix_df$col1 == "KEEP",]
netflix_df<-subset(netflix_df,select=-c(col1))

netflix_df <- inner_join(netflix_df,netflix_titles,by=c('MovieID'))
Movie_ratings <- netflix_df %>% group_by(MovieID, Title, Rating) %>% summarise(Count_of_Ratings = n())
```

```
## `summarise()` has grouped output by 'MovieID', 'Title'. You can override using the `.groups` argumen
```

```r
head(netflix_df)
```

```
##   CustomerID Rating       Date MovieID YearOfRelease                     Title
## 1    2059652      4 2005-09-05       2          2004 Isle of Man TT 2004 Review
## 2    1666394      3 2005-04-19       2          2004 Isle of Man TT 2004 Review
## 3    1759415      4 2005-04-22       2          2004 Isle of Man TT 2004 Review
## 4    1959936      5 2005-11-21       2          2004 Isle of Man TT 2004 Review
## 5     998862      4 2004-11-13       2          2004 Isle of Man TT 2004 Review
## 6    2625420      2 2004-12-06       2          2004 Isle of Man TT 2004 Review
```

```r
netflix_df <- na.omit(netflix_df)

netflix_df_5k <- head(netflix_df,5000)

summary(netflix_df)
```

```
##   CustomerID            Rating        Date              MovieID
##  Length:24053217    Min.   :1.0   Length:24053217    Length:24053217
##  Class :character   1st Qu.:3.0   Class :character   Class :character
##  Mode  :character   Median :4.0   Mode  :character   Mode  :character
##                     Mean   :3.6
##                     3rd Qu.:4.0
##                     Max.   :5.0
##  YearOfRelease         Title
##  Length:24053217    Length:24053217
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```r
netflix_data <-  lm(`MovieID` ~ CustomerID + Rating, data=netflix_df_5k)
```

Reading Movie Lense Data

```r
ml_movies_df=read.csv("C:/Users/anujt/Documents/R/GIT/dsc520/data/ml-latest-small/movies.csv",header=TRU
ml_ratings_df=read.csv("C:/Users/anujt/Documents/R/GIT/dsc520/data/ml-latest-small/ratings.csv",header=
ml_tags_df=read.csv("C:/Users/anujt/Documents/R/GIT/dsc520/data/ml-latest-small/tags.csv",header=TRUE)
ml_df <- inner_join(ml_ratings_df,ml_movies_df, by=c("movieId"))
ml_df <- subset(ml_df,select=-c(timestamp))
summary(ml_df)
```

```
##      userId         movieId           rating         title
##  Min.   :  1.0   Min.   :     1   Min.   :0.500   Length:100836
##  1st Qu.:177.0   1st Qu.:  1199   1st Qu.:3.000   Class :character
##  Median :325.0   Median :  2991   Median :3.500   Mode  :character
##  Mean   :326.1   Mean   : 19435   Mean   :3.502
##  3rd Qu.:477.0   3rd Qu.:  8122   3rd Qu.:4.000
##  Max.   :610.0   Max.   :193609   Max.   :5.000
```

```
##     genres
##  Length:100836
##  Class :character
##  Mode  :character
##
##
##
```

```
head(ml_df)
```

```
##   userId movieId rating                       title
## 1      1       1      4            Toy Story (1995)
## 2      1       3      4     Grumpier Old Men (1995)
## 3      1       6      4                 Heat (1995)
## 4      1      47      5 Seven (a.k.a. Se7en) (1995)
## 5      1      50      5  Usual Suspects, The (1995)
## 6      1      70      3  From Dusk Till Dawn (1996)
##                                        genres
## 1 Adventure|Animation|Children|Comedy|Fantasy
## 2                              Comedy|Romance
## 3                       Action|Crime|Thriller
## 4                            Mystery|Thriller
## 5                      Crime|Mystery|Thriller
## 6              Action|Comedy|Horror|Thriller
```

Implications

1. At best, recommendation systems serve both service providers and consumers alike. Consumers saves effort and time from going browsing through vast variety of products. Sellers drive sales and build loyality. However, recommendation systems also have unintended consequences.Recommendation Systems can result in biases. As consumers gets a humongous variety of options, they must pay great attention in evaluating potential products. Moreover, along with invested time lost, consumers can not unlist, unread or unwatch goods that turn out to be a poor fit.

2. "Do I like this?", "Should I like this?": Surprisingly, recommendation systems alter how much consumers are willing to pay for a product that they just saw. Consumers don't just prefer what they have experienced; they prefer what the system said they would like. This is surprising as consumers shouldn't need a system to tell them how much they enjoyed a product they just saw.

Limitations

1. Cold start problem: When building a new system, there would be no user data to start with. Only approach to be used in that situation is to first use content-based filtering first and then move on to collaborative filtering approach.

2. Scalability: As the number of users grow, the algorithms suffer scalability issues. If you have 10 million customers and 100,000 movies, you would have to create a sparse matrix with one trillion elements.

3. Lack of right data: Humans are not perfect at providing ratings, hence, input data may not always be accurate because .

Concluding Remarks

Recommendation systems provide recommendations by taking what other people recommend as well as our selections into account.

Although recommendation systems help both buyers and sellers greatly, they are not perfect and have certain limitations and implications.

Collaborative Filtering is a widely used and very effective solution for recommendation systems.

References

1. https://data-flair.training/blogs/data-science-projects-code/
2. https://rpubs.com/vsi/movielens
3. https://towardsdatascience.com/brief-on-recommender-systems-b86a1068a4dd
4. https://www.mygreatlearning.com/blog/masterclass-on-movie-recommendation-system/
5. https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781785884696/8/ch08lvl1sec62/limitations-of-a-recommendation-system