

Topic: Recommendation Engine

Q) Build a recommender system with the given data using UBCF.

Description of the data

In this dataset have users on the rows rated the jokes in the columns. The dataset comprises two csv files, Jokes.csv and Rating.csv is formatted as an excel file. The Jokes file consist of joke_id and Jokes & Rating.csv has the ratings given by the users to the jokes. Each rating is from (-10.00 to +10.00) and 99 corresponds to a null rating (user did not rate that joke). Note that the ratings are real values ranging from -10.00 to +10.00.

| | id | user_id | joke_id | Rating |
|----|-----------|---------|---------|--------|
| 1 | 31030_110 | 31030 | 110 | 2.750 |
| 2 | 16144_109 | 16144 | 109 | 5.094 |
| 3 | 23098_6 | 23098 | 6 | -6.438 |
| 4 | 14273_86 | 14273 | 86 | 4.406 |
| 5 | 18419_134 | 18419 | 134 | 9.375 |
| 6 | 5782_14 | 5782 | 14 | -1.781 |
| 7 | 34059_62 | 34059 | 62 | 6.250 |
| 8 | 1159_9 | 1159 | 9 | -2.906 |
| 9 | 17736_32 | 17736 | 32 | -5.750 |
| 10 | 22626_22 | 22626 | 22 | 1.656 |
| 11 | 6610_96 | 6610 | 96 | 1.656 |
| 12 | 18725_13 | 18725 | 13 | 0.688 |
| 13 | 23340_130 | 23340 | 130 | 1.312 |

Loading Required Packages and dataset:

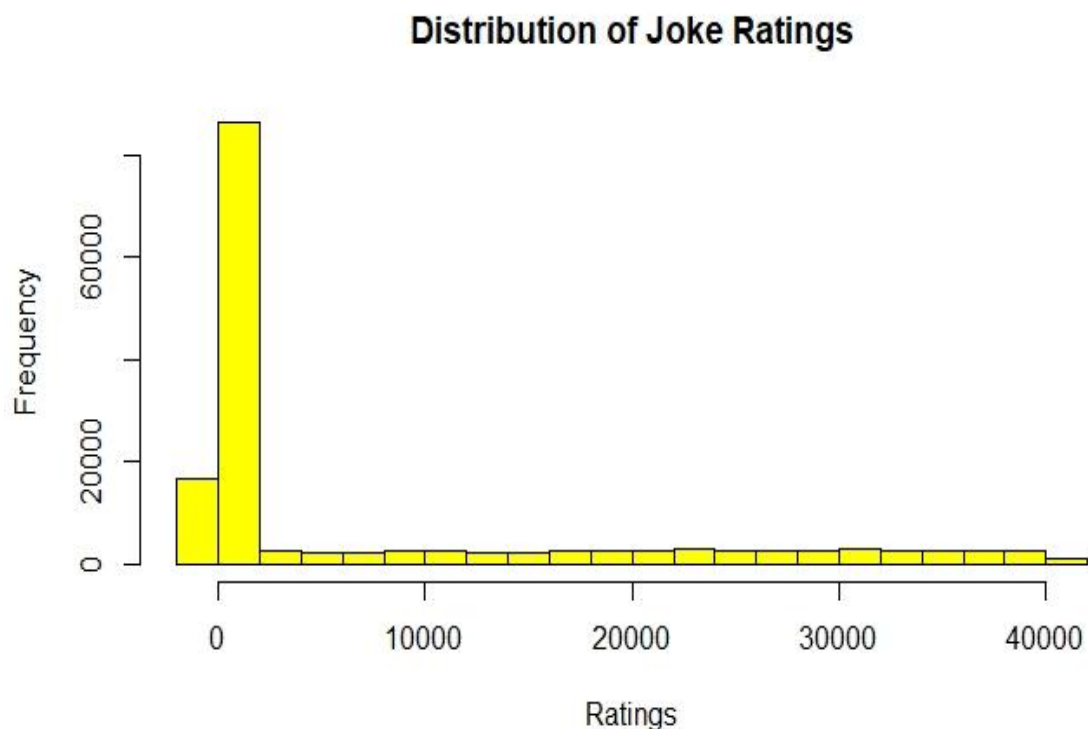
- 1) Installing necessary packages like “recommender lab” , “ caTools” & “reshape2)
- 2) Loading the dataset using the library readxl

Data Preprocessing:

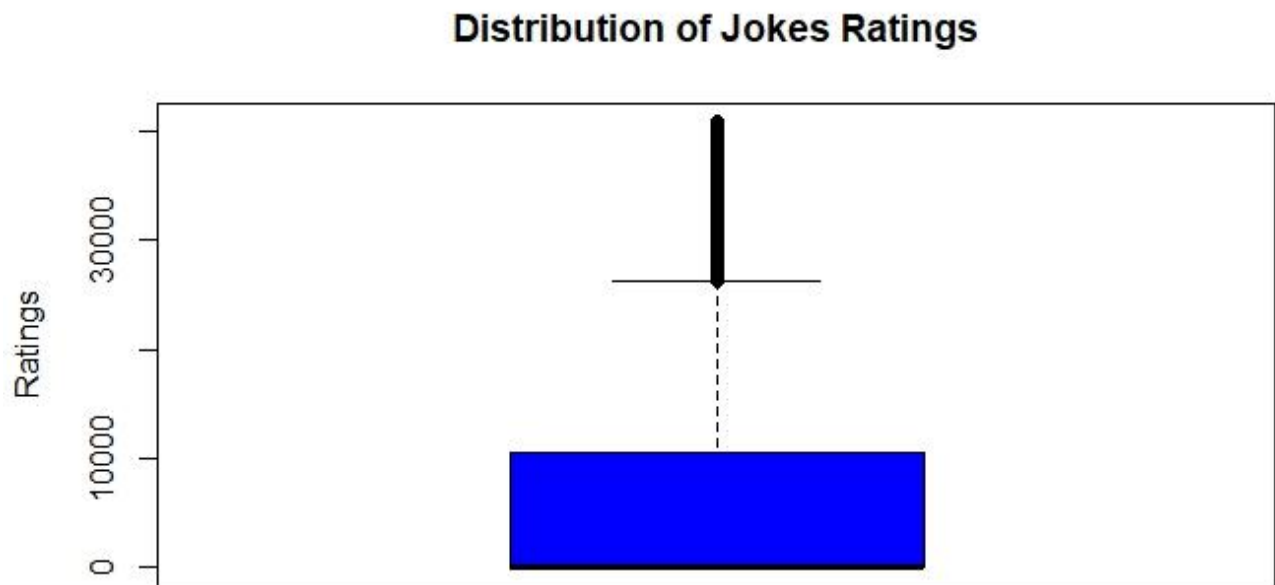
- 1) Removing unnecessary columns like “id”.
- 2) Checking the NA values, as there are no NA values no further imputation is required.
- 3) There are 50000 observations in the dataset with 3 variables after removing “id” column.

Graphical Representation of Ratings:

Histogram Representation:



BoxPlot Representation:



Converting Data into matrix format for further process:

- 1) Now, converting the data into matrix format after data preprocessing.
- 2) Now, the new dimensions of the matrix are [24175, 139].
- 3) Now based on the “recommender lab real rating matrix format”, object is created with the converted matrix.
- 4) Built 5 different recommender functions, using 5 different recommendations.
- 5) Here are the 5-recommender function: UBCF, IBFC, SVD, POPULAR and binarize functions.

Creating Recommendations for Users:

- 1) With the help of above constructed recommender system, now we can recommend jokes for each individual user.
- 2) No of recommendation taken are 2.

For User_id # 333

- With UBFC and IBFC, there are no recommendation for the user
- With SVD, the recommended jokes are “1” and “2”.
- With POPULAR, the recommended jokes are “79” and “104”.
- With binarize, the recommended jokes are “9” and “11”.

For User_id # 4

- With UBCF and IBFC, the recommended jokes are “119” and “79”.
- With IBCF, the recommended jokes are “8” and “18”.
- With SVD, the recommended jokes are “106” and “17”.
- With POPULAR, jokes recommended jokes are “79” and “104”.
- With binarize, the recommended jokes are “6” and “8”.

✓ Here 8 and 79 are repeated twice, so these jokes are highly recommended.

For User_id # 500

- With UBFC and IBFC, there are no recommendation for the user.
- With SVD, the recommended jokes are “1” and “2”.
- With POPULAR the recommended jokes are “79” and “104”.
- With binarize, the recommended jokes are “4” and “9”.

Hints:

1. Business Problem
 - 1.1. Objective
 - 1.2. Constraints (if any)
2. Data Pre-processing
 - 2.1 Data cleaning, Feature Engineering, EDA etc.
3. Model Building
 - 3.1 Partition the dataset
 - 3.2 Model(s) - Reasons to choose any algorithm
 - 3.3 Model(s) Improvement steps
 - 3.4 Model Evaluation
 - 3.5 Python and R codes
4. Deployment
 - 4.1 Deploy solutions using R shiny and Python Flask.
5. Result Share the benefits/impact of the solution - how or in what way the business (client) gets benefit from the solution provided.

Note:

1. For each assignment the solution should be submitted in the format
2. Research and Perform all possible steps for improving the model(s) recommendations
3. All the codes (executable programs) are running without errors
4. Documentation of the module should be submitted along with R & Python codes, elaborating on every step mentioned here