_ 2

June 30, 2021

1 Anime Recommendations Database : TASK

1.Of all anime having at least 1000 ratings, which anime has the maximum average rating ? anime_id = 28977

2.How many anime with atleast 1000 ratings have an average rating greater than 9 ?

3.Which is the most watched anime i.e. the anime rated by most number of users ?

4.What are the top three recommendations for the user with user_id 8086 ?

5.List top three users whom you would recommend the anime with anime_id 4935 ?

[1]:
```
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
import numpy as np
import os
import pandas as pd
import seaborn as sns
from datetime import datetime
import tensorflow as tf
```

[2]:
```
rating = pd.read_csv("rating.csv")
rating.head()
```

[2]:

	user_id	anime_id	rating
0	1	20	-1
1	1	24	-1
2	1	79	-1
3	1	226	-1
4	1	241	-1

[3]:
```
anime = pd.read_csv("anime.csv")
anime.head()
```

[3]:

	anime_id	name	\
0	32281	Kimi no Na wa.	
1	5114	Fullmetal Alchemist: Brotherhood	
2	28977	Gintama°	
3	9253	Steins;Gate	

```
4      9969                      Gintama&#039;;

                                                  genre   type episodes  rating  \
0               Drama, Romance, School, Supernatural  Movie        1    9.37
1  Action, Adventure, Drama, Fantasy, Magic, Mili…     TV       64    9.26
2  Action, Comedy, Historical, Parody, Samurai, S…     TV       51    9.25
3                                  Sci-Fi, Thriller     TV       24    9.17
4  Action, Comedy, Historical, Parody, Samurai, S…     TV       51    9.16

   members
0   200630
1   793665
2   114262
3   673572
4   151266
```

[5]:
```
rating.describe()
```

[5]:
```
            user_id      anime_id        rating
count  7.813737e+06  7.813737e+06  7.813737e+06
mean   3.672796e+04  8.909072e+03  6.144030e+00
std    2.099795e+04  8.883950e+03  3.727800e+00
min    1.000000e+00  1.000000e+00 -1.000000e+00
25%    1.897400e+04  1.240000e+03  6.000000e+00
50%    3.679100e+04  6.213000e+03  7.000000e+00
75%    5.475700e+04  1.409300e+04  9.000000e+00
max    7.351600e+04  3.451900e+04  1.000000e+01
```

[6]:
```
#  NA
data1 = rating.dropna()
data2 = anime.dropna()
```

[6]:
```
(12017, 7)
```

[8]:
```
#
anime_fulldata=pd.merge(anime,rating,on='anime_id',suffixes= ['', '_user'])
anime_fulldata = anime_fulldata.rename(columns={'name': 'anime_title',␣
 ↪'rating_user': 'user_rating'})
anime_fulldata.head()
```

[8]:
```
   anime_id     anime_title                                  genre   type  \
0     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
1     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
2     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
3     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
4     32281  Kimi no Na wa.  Drama, Romance, School, Supernatural  Movie
```

	episodes	rating	members	user_id	user_rating
0	1	9.37	200630	99	5
1	1	9.37	200630	152	10
2	1	9.37	200630	244	10
3	1	9.37	200630	271	10
4	1	9.37	200630	278	-1

2 1.Maximum average rating ? anime_id = 28977?

Answer: name : Gintama°

[4]:
```
C = anime['rating'].mean()
m = anime['members'].quantile(0.85)
q_animes = anime.copy().loc[anime['members'] >= m]
q_animes.shape
```

[4]: (1844, 7)

[8]:
```
def weighted_rating(x, m=m, C=C):
    v = x['members']
    R = x['rating']
    # Calculation based on the IMDB formula
    return (v/(v+m) * R) + (m/(m+v) * C)
```

[9]:
```
q_animes['rating'] = q_animes.apply(weighted_rating, axis=1)
```

[10]:
```
#
q_animes = q_animes.sort_values('rating', ascending=False)
q_animes[['name', 'members', 'rating']].head(15)
```

[10]:

	name	members	rating
1	Fullmetal Alchemist: Brotherhood	793665	9.176491
3	Steins;Gate	673572	9.075286
0	Kimi no Na wa.	200630	9.054553
6	Hunter x Hunter (2011)	425855	8.985370
10	Clannad: After Story	456749	8.928221
13	Code Geass: Hangyaku no Lelouch R2	572888	8.877123
12	Gintama	336376	8.865627
15	Sen to Chihiro no Kamikakushi	466254	8.807269
4	Gintama'	151266	8.785268
16	Shigatsu wa Kimi no Uso	416397	8.783948
2	Gintama°	114262	8.759452
19	Code Geass: Hangyaku no Lelouch	715151	8.751883
23	One Punch Man	552458	8.720281
22	Cowboy Bebop	486824	8.707482
29	Tengen Toppa Gurren Lagann	562962	8.683734

[11]:
```
anime.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12294 entries, 0 to 12293
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   anime_id  12294 non-null  int64
 1   name      12294 non-null  object
 2   genre     12232 non-null  object
 3   type      12269 non-null  object
 4   episodes  12294 non-null  object
 5   rating    12064 non-null  float64
 6   members   12294 non-null  int64
dtypes: float64(1), int64(2), object(4)
memory usage: 672.5+ KB
```

[12]:
```
# NA for anime
anime.isnull().sum()
```

[12]:
```
anime_id      0
name          0
genre        62
type         25
episodes      0
rating      230
members       0
dtype: int64
```

[13]:
```
#
#anime_id = 28977
anime_id_28977  = anime[anime['anime_id']== 28977]
anime_id_28977
```

[13]:
```
   anime_id      name                                                genre type  \
2     28977  Gintama°  Action, Comedy, Historical, Parody, Samurai, S…   TV

  episodes  rating  members
2       51    9.25   114262
```

3 2.Average rating greater than 9 ?

Answer: Fullmetal Alchemist: Brotherhood Steins;Gate Kimi no Na wa 3

[14]:
```
q_animes_9 = q_animes[q_animes["rating"]>=9]
q_animes_9
```

[14]:
```
   anime_id                              name  \
1      5114  Fullmetal Alchemist: Brotherhood
3      9253                      Steins;Gate
0     32281                    Kimi no Na wa.

                                          genre   type episodes  \
1  Action, Adventure, Drama, Fantasy, Magic, Mili…     TV       64
3                                Sci-Fi, Thriller     TV       24
0             Drama, Romance, School, Supernatural  Movie        1

     rating  members
1  9.176491   793665
3  9.075286   673572
0  9.054553   200630
```

4 3. The most watched anime = Most number of users

Answer: Death Note

[15]:
```
max(anime["members"])
```

[15]:
```
1013917
```

[16]:
```
members =  anime[anime['members']== 1013917]
members
```

[16]:
```
    anime_id        name                                              genre  \
40      1535  Death Note  Mystery, Police, Psychological, Supernatural, …

   type episodes  rating  members
40   TV       37    8.71  1013917
```

[17]:
```
##   dataframe
name1 = anime[["anime_id","name"]]
members1 = anime[["anime_id","members"]]
merge = pd.merge(name1 , members1)
merge.sort_values(by=['members'], ascending=False)
```

[17]:
```
       anime_id                                        name  members
40         1535                                  Death Note  1013917
86        16498                              Shingeki no Kyojin   896229
804       11757                               Sword Art Online   893100
1          5114                Fullmetal Alchemist: Brotherhood   793665
159        6547                                   Angel Beats!   717796
…           …                                             …       …
10464     33662  Taka no Tsume 8: Yoshida-kun no X-Files       13
10424     33320                                  Suijun Genten       13
```

```
10444     34490                                      Sushi Azarashi       12
10990     34485                                          Ganko-chan       11
10997     34527  Gou-chan. Moko to Chinjuu no Mori no Nakama-tachi        5

[12294 rows x 3 columns]
```

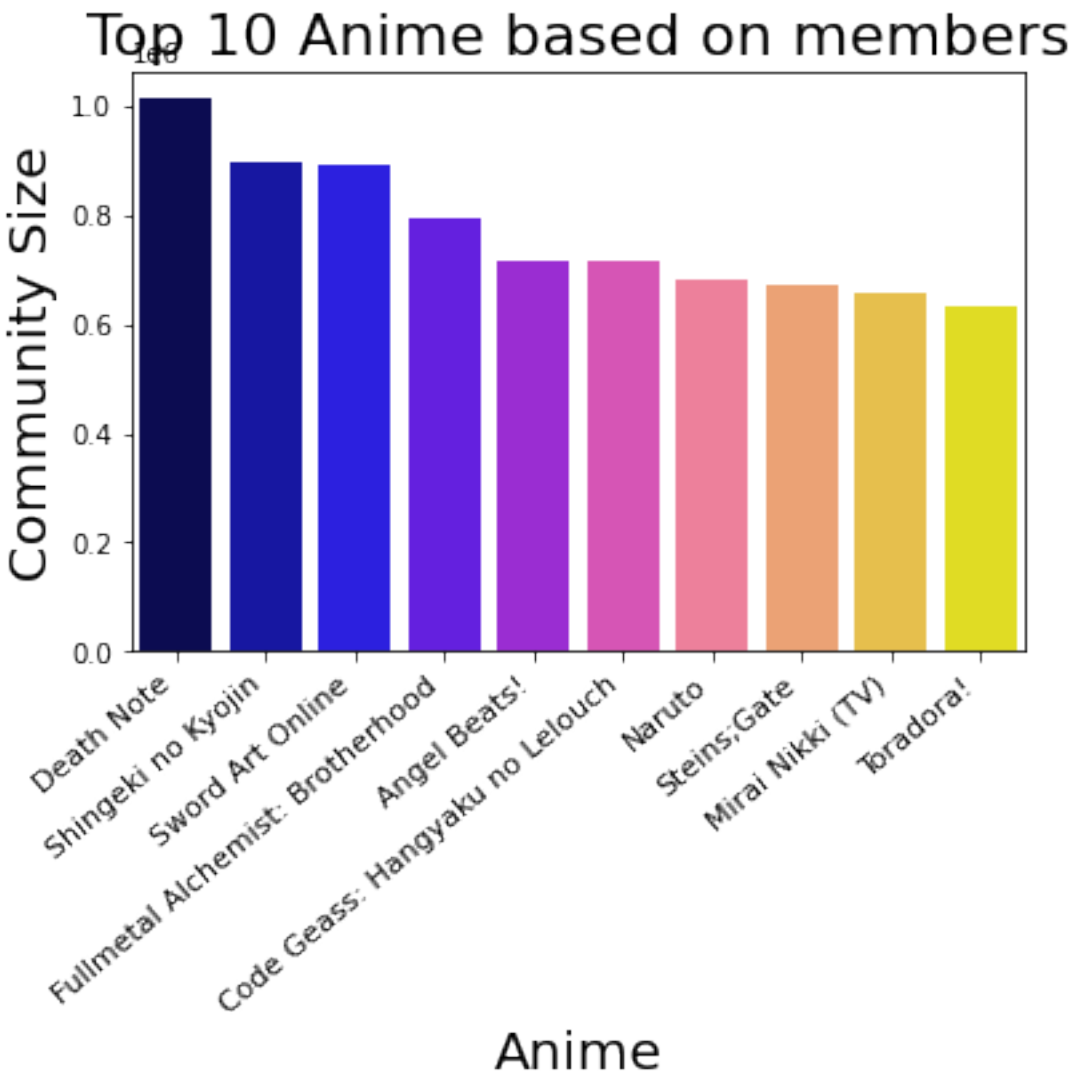
[18]:
```
anime.sort_values(by=['members'], ascending=False)
```

[18]:
```
       anime_id                                              name  \
40         1535                                        Death Note
86        16498                                    Shingeki no Kyojin
804       11757                                     Sword Art Online
1          5114                        Fullmetal Alchemist: Brotherhood
159        6547                                      Angel Beats!
…           …                                                 …
10464     33662           Taka no Tsume 8: Yoshida-kun no X-Files
10424     33320                                      Suijun Genten
10444     34490                                     Sushi Azarashi
10990     34485                                         Ganko-chan
10997     34527  Gou-chan. Moko to Chinjuu no Mori no Nakama-tachi

                                                  genre   type episodes  \
40     Mystery, Police, Psychological, Supernatural, …     TV       37
86          Action, Drama, Fantasy, Shounen, Super Power     TV       25
804            Action, Adventure, Fantasy, Game, Romance     TV       25
1      Action, Adventure, Drama, Fantasy, Magic, Mili…     TV       64
159          Action, Comedy, Drama, School, Supernatural     TV       13
…                                                    …    …       …
10464                                  Comedy, Parody  Movie        1
10424                                             NaN  Movie        1
10444                                          Comedy     TV       30
10990                                             NaN    NaN  Unknown
10997                                 Adventure, Kids  Movie        1

       rating  members
40       8.71  1013917
86       8.54   896229
804      7.83   893100
1        9.26   793665
159      8.39   717796
…         …       …
10464   10.00       13
10424    7.00       13
10444    3.00       12
10990     NaN       11
10997     NaN        5
```

```
[12294 rows x 7 columns]
```

[9]:
```
duplicate_anime=anime_fulldata.copy()
duplicate_anime.drop_duplicates(subset ="anime_title", keep = 'first', inplace
 ↪= True)

top10_animemembers=duplicate_anime[['anime_title', 'members']].sort_values(by =
 ↪'members',ascending = False).head(10)
ax=sns.barplot(x="anime_title", y="members", data=top10_animemembers,
 ↪palette="gnuplot2")
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11, rotation=40, ha="right")
ax.set_title('Top 10 Anime based on members',fontsize = 22)
ax.set_xlabel('Anime',fontsize = 20)
ax.set_ylabel('Community Size', fontsize = 20)
```

[9]:
```
Text(0, 0.5, 'Community Size')
```



5 4.The top three recommendations for the user with user_id 8086

Answer :Sen to Chihiro no Kamikakushi Mononoke Hime Howl no Ugoku Shiro

[29]:
```
rating.head(10)
```

[29]:

	user_id	anime_id	rating
0	1	20	-1
1	1	24	-1
2	1	79	-1
3	1	226	-1
4	1	241	-1
5	1	355	-1
6	1	356	-1
7	1	442	-1
8	1	487	-1
9	1	846	-1

[30]:
```
merge = pd.merge(anime,rating.drop("rating",axis=1))
merge
```

[30]:

	anime_id	name	\
0	32281	Kimi no Na wa.	
1	32281	Kimi no Na wa.	
2	32281	Kimi no Na wa.	
3	32281	Kimi no Na wa.	
4	32281	Kimi no Na wa.	
…	…	…	
7813722	6133	Violence Gekiga Shin David no Hoshi: Inma Dens…	
7813723	6133	Violence Gekiga Shin David no Hoshi: Inma Dens…	
7813724	6133	Violence Gekiga Shin David no Hoshi: Inma Dens…	
7813725	26081	Yasuji no Pornorama: Yacchimae!!	
7813726	26081	Yasuji no Pornorama: Yacchimae!!	

	genre	type	episodes	rating	\
0	Drama, Romance, School, Supernatural	Movie	1	9.37	
1	Drama, Romance, School, Supernatural	Movie	1	9.37	
2	Drama, Romance, School, Supernatural	Movie	1	9.37	
3	Drama, Romance, School, Supernatural	Movie	1	9.37	
4	Drama, Romance, School, Supernatural	Movie	1	9.37	
…	…	…	…	…	
7813722	Hentai	OVA	1	4.98	
7813723	Hentai	OVA	1	4.98	

```
7813724                                Hentai    OVA        1    4.98
7813725                                Hentai  Movie        1    5.46
7813726                                Hentai  Movie        1    5.46

         members  user_id
0         200630       99
1         200630      152
2         200630      244
3         200630      271
4         200630      278
…            …        …
7813722      175    39532
7813723      175    48766
7813724      175    60365
7813725      142    27364
7813726      142    48766

[7813727 rows x 8 columns]
```

[31]:
```
user_id_8086 = merge[merge['user_id']== 8086]
user_id_8086.sort_values(by=['rating'], ascending=False)
```

[31]:
```
         anime_id                            name  \
120550        199  Sen to Chihiro no Kamikakushi
228992        164                  Mononoke Hime
307287        431            Howl no Ugoku Shiro
702364        205               Samurai Champloo
750644        523               Tonari no Totoro
…             …                              …
6830698      1243             Night Head Genesis
6951665      2795       Dragonaut: The Resonance
7756014      2148                  Okane ga Nai
7370910      1734       Ajimu: Kaigan Monogatari
7420397      8449                  Togainu no Chi

                                                    genre   type episodes  \
120550                      Adventure, Drama, Supernatural  Movie        1
228992                          Action, Adventure, Fantasy  Movie        1
307287                   Adventure, Drama, Fantasy, Romance  Movie        1
702364   Action, Adventure, Comedy, Historical, Samurai…     TV       26
750644                     Adventure, Comedy, Supernatural  Movie        1
…                                                      …    …       …
6830698  Drama, Horror, Mystery, Psychological, Superna…     TV       24
6951665     Action, Drama, Fantasy, Mecha, Romance, Sci-Fi     TV       25
7756014                                Drama, Romance, Yaoi    OVA        4
7370910                     Comedy, Drama, Romance, School    ONA        4
7420397                         Action, Sci-Fi, Shounen Ai     TV       12
```

```
         rating  members  user_id
120550     8.93   466254     8086
228992     8.81   339556     8086
307287     8.74   333186     8086
702364     8.50   390076     8086
750644     8.48   271484     8086
…           …       …        …
6830698    6.88    20856     8086
6951665    6.81    45265     8086
7756014    6.58    27367     8086
7370910    6.48     9102     8086
7420397    6.42    53377     8086

[77 rows x 8 columns]
```

6 5.List top three users recommend the anime_id 4935

Answer :

[32]:
```
merge
```

[32]:
```
         anime_id                                                name  \
0           32281                                     Kimi no Na wa.
1           32281                                     Kimi no Na wa.
2           32281                                     Kimi no Na wa.
3           32281                                     Kimi no Na wa.
4           32281                                     Kimi no Na wa.
…             …                                                 …
7813722      6133  Violence Gekiga Shin David no Hoshi: Inma Dens…
7813723      6133  Violence Gekiga Shin David no Hoshi: Inma Dens…
7813724      6133  Violence Gekiga Shin David no Hoshi: Inma Dens…
7813725     26081                   Yasuji no Pornorama: Yacchimae!!
7813726     26081                   Yasuji no Pornorama: Yacchimae!!

                                      genre   type episodes  rating  \
0        Drama, Romance, School, Supernatural  Movie        1    9.37
1        Drama, Romance, School, Supernatural  Movie        1    9.37
2        Drama, Romance, School, Supernatural  Movie        1    9.37
3        Drama, Romance, School, Supernatural  Movie        1    9.37
4        Drama, Romance, School, Supernatural  Movie        1    9.37
…                                          …     …       …      …
7813722                              Hentai    OVA        1    4.98
7813723                              Hentai    OVA        1    4.98
7813724                              Hentai    OVA        1    4.98
7813725                              Hentai  Movie        1    5.46
7813726                              Hentai  Movie        1    5.46
```

```
         members  user_id
0         200630       99
1         200630      152
2         200630      244
3         200630      271
4         200630      278
…            …        …
7813722      175    39532
7813723      175    48766
7813724      175    60365
7813725      142    27364
7813726      142    48766

[7813727 rows x 8 columns]
```

[34]:
```
anime_id_4935 = merge[merge['anime_id']==4935]
anime_id_4935.sort_values(by=['rating'], ascending=False)
```

[34]:
```
         anime_id        name                       genre type episodes  rating  \
6482564      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482575      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482583      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482582      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482581      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482580      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482579      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482578      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482577      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482576      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482574      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482565      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482573      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482572      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482571      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482570      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482569      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482568      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482567      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482566      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05
6482584      4935  Ikkyuu-san  Comedy, Historical, Kids   TV      296    7.05

         members  user_id
6482564      720     1822
6482575      720    48766
6482583      720    64820
6482582      720    58378
```

```
6482581      720    56650
6482580      720    55670
6482579      720    53060
6482578      720    50822
6482577      720    50656
6482576      720    50537
6482574      720    39111
6482565      720     7000
6482573      720    36575
6482572      720    30597
6482571      720    24931
6482570      720    24408
6482569      720    18944
6482568      720    15448
6482567      720    13539
6482566      720    12725
6482584      720    65855
```

[ ]:
```
##
```

[ ]:
```
nonull_anime=anime_fulldata.copy()
nonull_anime.dropna(inplace=True)
from collections import defaultdict

all_genres = defaultdict(int)

for genres in nonull_anime['genre']:
    for genre in genres.split(','):
        all_genres[genre.strip()] += 1

from wordcloud import WordCloud

genres_cloud = WordCloud(width=800, height=400, background_color='white',␣
 ↪colormap='gnuplot').generate_from_frequencies(all_genres)
plt.imshow(genres_cloud, interpolation='bilinear')
plt.axis('off')
```

Big Data Analysis FINAL2

7109018022

data source : https://www.kaggle.com/CooperUnion/anime-recommendations-database (https://www.kaggle.com/CooperUnion/anime-recommendations-database)

Load package

```
library(magrittr) #%>%
library(data.table) #fread
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(sparklyr)
```

```
##
## Attaching package: 'sparklyr'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(devtools) #sinkr
```

```
## Loading required package: usethis
```

```
library(sinkr) #iterative PCA
```

data

```
data = fread("./anime.csv", stringsAsFactors = T, encoding = "UTF-8")
head(data)
```

clean data

```
data$genre = as.character(data$genre)
n = unlist(count(data)) #number of column

for ( i in 1:n ){
  a = as.vector(unlist(strsplit(data[i]$genre,"[,]")))
  if ( length(a) > 1 ){
    data[i]$genre = a[1]
    for ( j in 2:length(a) ){
      b = data[i,]
      b$genre = a[j]
      data = rbind(data,b)
    }
  }
}
data$genre = as.factor(data$genre)
```

examine NULL data

```
sum(is.na(data))
```

```
## [1] 705
```

```
sum(is.na(data$anime_id)) #Anime ID
```

```
## [1] 0
```

```
sum(is.na(data$name)) #Anime NAME
```

```
## [1] 0
```

```
sum(is.na(data$genre)) #Type of animation (separated by commas)
```

```
## [1] 0
```

```
sum(is.na(data$type)) #OVA, movies, TV...
```

```
## [1] 0
```

```
sum(is.na(data$episodes)) #series
```

```
## [1] 0
```

```
sum(is.na(data$rating))
```

```
## [1] 705
```

```
sum(is.na(data$members)) #A few people commented
```

```
## [1] 0
```

deal with missing value

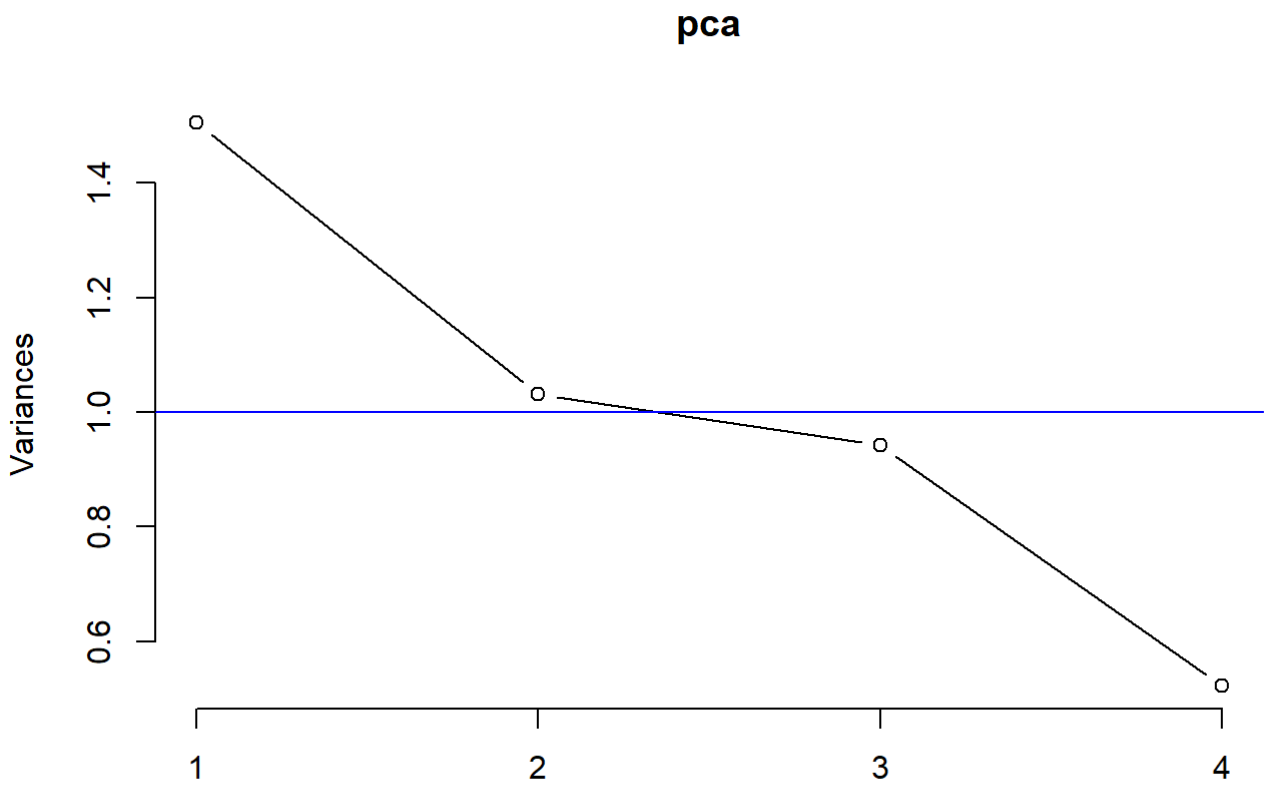
PCA

```
d = data %>%  arrange(anime_id) %>%
  mutate(across(everything() , as.numeric))  #convert to numeric
pca = prcomp(formula = ~ genre + type + episodes + members , data = d , scale = TRUE)
pca
```

```
## Standard deviations (1, .., p=4):
## [1] 1.2266650 1.0152040 0.9709273 0.7224639
##
## Rotation (n x k) = (4 x 4):
##                 PC1        PC2         PC3        PC4
## genre    -0.1591208  0.7006178 -0.69540461 -0.0150916
## type      0.6918131  0.1077894 -0.06513177  0.7110091
## episodes  0.6158393  0.3871595  0.26290216 -0.6338233
## members   0.3417852 -0.5895963 -0.66562206 -0.3041486
```

Scree plot

```
plot(pca , type = "line" )
abline(h=1, col="blue")
```



Thus, select Q = 2

iterative PCA

```
Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jdk1.8.0_111\\jre') #Positioning JAVA
sc <- spark_connect(master="local") #Which host SPARK to choose

spark.df = sdf_copy_to(sc, d , overwrite = TRUE)  # convert R_dataFrame to spark_sql_dataFram
e
x = spark.df %>%
  sdf_collect() %>% as.data.frame() %>% as.matrix()
xa = dineof(x,2)$Xa
```

```
## [1] "1 EOF ; RMS = 17773.61801909"
## [1] "1 EOF ; RMS = 17665.25252343"
## [1] "1 EOF ; RMS = 17565.74142401"
## [1] "1 EOF ; RMS = 17474.71856283"
## [1] "1 EOF ; RMS = 17392.22353828"
## [1] "1 EOF ; RMS = 17318.28847318"
## [1] "1 EOF ; RMS = 17252.93365836"
## [1] "1 EOF ; RMS = 17196.16716783"
## [1] "1 EOF ; RMS = 17147.98459424"
## [1] "1 EOF ; RMS = 17108.36887285"
## [1] "1 EOF ; RMS = 17077.29019722"
## [1] "1 EOF ; RMS = 17054.70602911"
## [1] "1 EOF ; RMS = 17040.561203"
## [1] "1 EOF ; RMS = 17034.78812447"
## [1] "1 EOF ; RMS = 17037.30705966"
## [1] "2 EOF ; RMS = 16807.89312142"
```

```
X = as.data.frame(xa)
head(X)
```

```
sum(is.na(X))
```

```
## [1] 0
```

```
spark_disconnect(sc)
```

regression analysis

Elastic net model

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-1
```

```
##
## Attaching package: 'glmnet'
```

```
## The following object is masked from 'package:sparklyr':
##
##     na.replace
```

```
train = sample(c(T,T,T,T,F),nrow(data),rep=T) ##80/20 train/test split
test = (!train)

y = X %>% select(rating) %>% as.matrix()
v = X %>% select(genre,type,episodes,members) %>% as.matrix()

set.seed(42)
cv_10 = trainControl(method = "cv", number = 10)
```

model

```
hit_elnet = train(
  rating ~ genre + type + episodes + members, data = X[train,] ,
  method = "glmnet",
  trControl = cv_10
)
hit_elnet
```

```
## glmnet
##
## 29155 samples
##     4 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 26240, 26238, 26241, 26240, 26238, 26240, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda        RMSE       Rsquared   MAE
##   0.10   0.0007731472  0.8999649  0.1878578  0.6817631
##   0.10   0.0077314723  0.8999680  0.1878574  0.6817669
##   0.10   0.0773147235  0.9006669  0.1878027  0.6823413
##   0.55   0.0007731472  0.8999659  0.1878584  0.6817323
##   0.55   0.0077314723  0.8999987  0.1878513  0.6817073
##   0.55   0.0773147235  0.9036225  0.1864616  0.6838222
##   1.00   0.0007731472  0.8999666  0.1878581  0.6817227
##   1.00   0.0077314723  0.9000573  0.1878255  0.6816676
##   1.00   0.0773147235  0.9096138  0.1809713  0.6876321
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0.1 and lambda = 0.0007731472.
```

expanding the feature space

```
hit_elnet_int = train(
  rating ~ (genre + type + episodes + members)^2 , data = X[train,] ,
  method = "glmnet",
  trControl = cv_10,
  tuneLength = 10
)
hit_elnet_int
```

```
## glmnet
##
## 29155 samples
##     4 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 26240, 26239, 26241, 26239, 26240, 26239, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda        RMSE       Rsquared   MAE
##   0.1    0.0001786070  0.8745166  0.2337601  0.6565284
##   0.1    0.0004126053  0.8745187  0.2337559  0.6565406
##   0.1    0.0009531719  0.8746020  0.2336107  0.6568233
##   0.1    0.0022019507  0.8749910  0.2329954  0.6575869
##   0.1    0.0050867918  0.8764559  0.2306831  0.6595454
##   0.1    0.0117511490  0.8804287  0.2240974  0.6635243
##   0.1    0.0271466791  0.8873152  0.2120847  0.6699785
##   0.1    0.0627123512  0.8950838  0.1983244  0.6768566
##   0.1    0.1448736687  0.8990906  0.1930080  0.6798658
##   0.1    0.3346769731  0.9053907  0.1879397  0.6850618
##   0.2    0.0001786070  0.8745047  0.2337882  0.6564408
##   0.2    0.0004126053  0.8745201  0.2337552  0.6565496
##   0.2    0.0009531719  0.8746019  0.2336138  0.6568419
##   0.2    0.0022019507  0.8749922  0.2330006  0.6576425
##   0.2    0.0050867918  0.8765228  0.2305960  0.6597453
##   0.2    0.0117511490  0.8809254  0.2232965  0.6641975
##   0.2    0.0271466791  0.8889320  0.2092942  0.6714164
##   0.2    0.0627123512  0.8971254  0.1950580  0.6782132
##   0.2    0.1448736687  0.9004661  0.1926895  0.6806785
##   0.2    0.3346769731  0.9114706  0.1854951  0.6895726
##   0.3    0.0001786070  0.8745003  0.2337946  0.6564410
##   0.3    0.0004126053  0.8745143  0.2337642  0.6565601
##   0.3    0.0009531719  0.8745968  0.2336229  0.6568666
##   0.3    0.0022019507  0.8749938  0.2330034  0.6577104
##   0.3    0.0050867918  0.8766118  0.2304684  0.6599736
##   0.3    0.0117511490  0.8813754  0.2225949  0.6646309
##   0.3    0.0271466791  0.8909575  0.2057050  0.6732545
##   0.3    0.0627123512  0.8984038  0.1932670  0.6789306
##   0.3    0.1448736687  0.9022999  0.1920848  0.6819505
##   0.3    0.3346769731  0.9200750  0.1788169  0.6962549
##   0.4    0.0001786070  0.8745016  0.2337932  0.6564496
##   0.4    0.0004126053  0.8745160  0.2337621  0.6565742
##   0.4    0.0009531719  0.8745999  0.2336201  0.6568951
##   0.4    0.0022019507  0.8750068  0.2329878  0.6577847
##   0.4    0.0050867918  0.8767292  0.2302892  0.6602225
##   0.4    0.0117511490  0.8819130  0.2217351  0.6651088
##   0.4    0.0271466791  0.8933274  0.2014372  0.6752738
##   0.4    0.0627123512  0.8988868  0.1930598  0.6792492
##   0.4    0.1448736687  0.9047180  0.1907488  0.6838049
##   0.4    0.3346769731  0.9309866  0.1649028  0.7050549
##   0.5    0.0001786070  0.8745044  0.2337893  0.6564820
##   0.5    0.0004126053  0.8745218  0.2337530  0.6566173
##   0.5    0.0009531719  0.8746059  0.2336095  0.6569318
##   0.5    0.0022019507  0.8750243  0.2329569  0.6578668
##   0.5    0.0050867918  0.8768148  0.2301577  0.6604123
##   0.5    0.0117511490  0.8825398  0.2206926  0.6656668
```

```
##   0.5    0.0271466791  0.8961242  0.1963241  0.6775972
##   0.5    0.0627123512  0.8994721  0.1927732  0.6796581
##   0.5    0.1448736687  0.9077247  0.1885078  0.6861249
##   0.5    0.3346769731  0.9405197  0.1528862  0.7129446
##   0.6    0.0001786070  0.8745062  0.2337854  0.6564946
##   0.6    0.0004126053  0.8745234  0.2337515  0.6566349
##   0.6    0.0009531719  0.8746108  0.2336040  0.6569702
##   0.6    0.0022019507  0.8750450  0.2329263  0.6579467
##   0.6    0.0050867918  0.8768739  0.2300880  0.6604769
##   0.6    0.0117511490  0.8832752  0.2194634  0.6663284
##   0.6    0.0271466791  0.8968066  0.1953477  0.6779633
##   0.6    0.0627123512  0.9001756  0.1923591  0.6801739
##   0.6    0.1448736687  0.9113617  0.1849701  0.6888798
##   0.6    0.3346769731  0.9469234  0.1510670  0.7191770
##   0.7    0.0001786070  0.8745060  0.2337873  0.6565046
##   0.7    0.0004126053  0.8745240  0.2337510  0.6566520
##   0.7    0.0009531719  0.8746151  0.2335979  0.6570051
##   0.7    0.0022019507  0.8750708  0.2328862  0.6580308
##   0.7    0.0050867918  0.8769406  0.2300050  0.6605375
##   0.7    0.0117511490  0.8841426  0.2179850  0.6671152
##   0.7    0.0271466791  0.8975058  0.1943383  0.6783396
##   0.7    0.0627123512  0.9010035  0.1917874  0.6807928
##   0.7    0.1448736687  0.9156833  0.1795653  0.6921789
##   0.7    0.3346769731  0.9532148  0.1511370  0.7254736
##   0.8    0.0001786070  0.8745079  0.2337852  0.6565191
##   0.8    0.0004126053  0.8745283  0.2337446  0.6566749
##   0.8    0.0009531719  0.8746221  0.2335891  0.6570347
##   0.8    0.0022019507  0.8751016  0.2328366  0.6581204
##   0.8    0.0050867918  0.8770161  0.2299068  0.6606078
##   0.8    0.0117511490  0.8851502  0.2162424  0.6680392
##   0.8    0.0271466791  0.8981936  0.1933000  0.6787345
##   0.8    0.0627123512  0.9019630  0.1910239  0.6815153
##   0.8    0.1448736687  0.9205713  0.1721183  0.6959952
##   0.8    0.3346769731  0.9606051  0.1511370  0.7328377
##   0.9    0.0001786070  0.8745102  0.2337819  0.6565355
##   0.9    0.0004126053  0.8745310  0.2337409  0.6566953
##   0.9    0.0009531719  0.8746330  0.2335721  0.6570827
##   0.9    0.0022019507  0.8751337  0.2327845  0.6582120
##   0.9    0.0050867918  0.8771003  0.2297926  0.6606868
##   0.9    0.0117511490  0.8862756  0.2142755  0.6690233
##   0.9    0.0271466791  0.8985207  0.1929169  0.6789443
##   0.9    0.0627123512  0.9030640  0.1900157  0.6823510
##   0.9    0.1448736687  0.9245175  0.1665053  0.6990289
##   0.9    0.3346769731  0.9692318  0.1511370  0.7410538
##   1.0    0.0001786070  0.8745091  0.2337824  0.6565472
##   1.0    0.0004126053  0.8745336  0.2337372  0.6567142
##   1.0    0.0009531719  0.8746362  0.2335656  0.6571018
##   1.0    0.0022019507  0.8751609  0.2327419  0.6582928
##   1.0    0.0050867918  0.8771923  0.2296647  0.6607722
##   1.0    0.0117511490  0.8876078  0.2119121  0.6701721
##   1.0    0.0271466791  0.8987491  0.1927531  0.6791033
##   1.0    0.0627123512  0.9043165  0.1887139  0.6833192
##   1.0    0.1448736687  0.9280536  0.1619289  0.7017617
##   1.0    0.3346769731  0.9792971  0.1511370  0.7502285
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0.3 and lambda = 0.000178607.
```

select alpha and lambda

```
get_best_result = function(caret_fit) {
  best = which(rownames(caret_fit$results) == rownames(caret_fit$bestTune))
  best_result = caret_fit$results[best, ]
  rownames(best_result) = NULL
  best_result
}
get_best_result(hit_elnet_int)
```

MSE

```
glmnet.fit = glmnet(v[train,] , y[train,] , alpha = get_best_result(hit_elnet_int)[1] , lambd
a = get_best_result(hit_elnet_int)[2])
glmnet.pred = predict(glmnet.fit,s = get_best_result(hit_elnet_int)[2] , v[test,])
mean((glmnet.pred - X$rating[test])^2)
```

```
## [1] 0.8389254
```