Movie recommendation System

This project is about Recommending new movie after seing one movie.

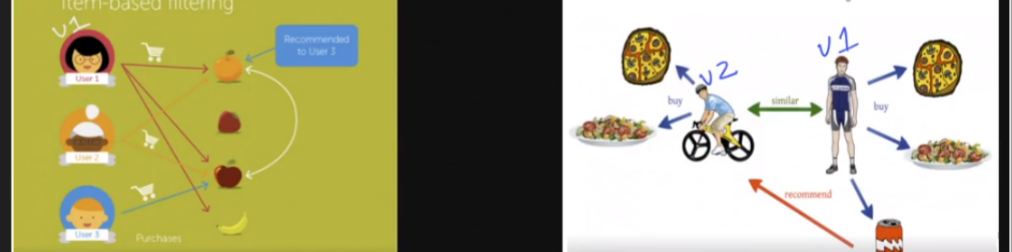
Recommendation system are of two types:

1)Content based

2)Collabrative filtering

In content based attributes are used for recommendation.

And in collaborative filtering human behaviour used for recommendation.



Dataset link-><https://grouplens.org/datasets/movielens/100k/>

Importing Libraries of Python

|  |  |
| --- | --- |
|  | import pandas as pd |
|  | import numpy as np |
|  | import warnings |
|  |  |
|  | warnings.filterwarnings('ignore') |

Warings is used as to avoid any waring.

Path of my file from where I have to start my work

1.I have taken u.data from ml-100k dataset.

read=pd.read\_csv("C:/Users/HP/Desktop/ml-100k/u.data",sep="\t",names=column\_name)

2. read.head()

Used to see the top 5 values

3.read.shape()

To know the shape (row,col)

4. column\_name=["user\_id","name\_id","rating","timestamp"]

This I used to give name to columns.

5.read[‘used\_id’].nunique()

For getting unique no. of users.

6.read[‘item\_id’].nuique()

For getting unique movies.

7.But right now we don’t know ‘item\_id’ of u.data corresponds to which movie. So,we have to import another file from the same directory.

read1=pd.read\_csv("C:/Users/HP/Desktop/ml-100k/u.item",sep="\|",header=None)

8. read\_data=read1[[0,1]]

To get only two columns from read1. Which contains->item\_id and name of movie which was required.

9.we have to merge data of (read,read\_data) using pandas library.

dg=pd.merge(read,read\_data,on="name\_id")

we merge it on “name\_id”

10.dg.tail()

To get last 5 values

Starting of exploratory data Analysis

11. import matplotlib.pyplot as plt

import seaborn as sns

12.dg.groupby('title').mean()['rating'].sort\_values(ascending=False)

Groupby ‘title’ with mean of all values and discard of all other values and only take rating column into consideration,and sort the answer in descending order to get maximum value at first.

13.dg.groupby('title').count()['rating'].sort\_values(ascending=False)

Groupby ‘title’ with count of all values and discard of all other values and only take rating column into consideration,and sort the answer in descending order to get maximum value at first.

14.rating=pd.DataFrame(dg.groupby('title').count()['rating'])

To create dataframe of same datas

15.rating['mean of rating']=pd.DataFrame(dg.groupby('title').mean()['rating'])

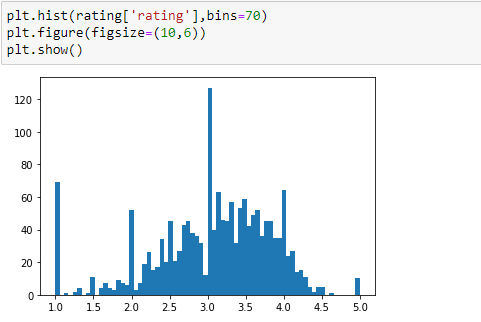
To create dataframe of second data by giving name to it as ‘mean of rating’.

16.plt.hist(rating["rating"],bins=50)

From the histogram we come to see that there are very less no. of person who have watched the movie and given rating 5.and most of the person have given rating as 3.

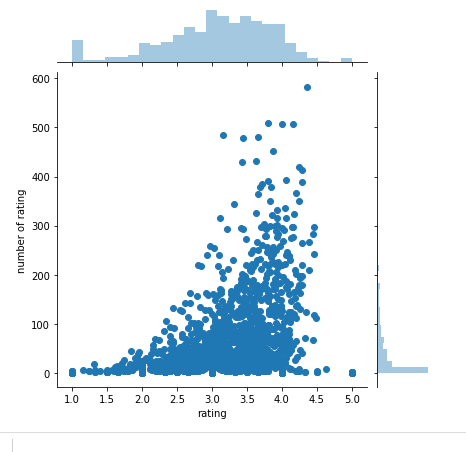
17.plt.hist(rating["count"],bins=50)

From this histogram we come to see the count of no. times movie is being seen by person.



18.sns.jointplot(x='rating',y='count',data=rating,kind='scatter',alpha=0.5)

We create jointplot to see the dots the data points and linear trend we are able to see.



19.movirate=dg.pivot\_table(index="user\_id",columns="title",values="rating")

We make table where row is user\_id,column is title and value is rating means from where data came from.

20.star\_wars\_user\_rating=movirate['Star Wars (1977)']

From movie matrix we pick one movie i.e star wars whicj has highest rating.

21.similar\_to\_starwars=movirate.corrwith(star\_wars\_user\_rating)

Use to correlate the starwrs with all other movie of movie matrix

22.convert\_corr\_of\_starwars=pd.DataFrame(similar\_to\_starwars,columns=['Correlation'])

convert\_corr\_of\_starwars.dropna(inplace=True)

inplace is used to permanent change of value.

Convert data to dataframe using pandas library and then drop NA value

23.we only recommend that movie which have high correlation value

convert\_corr\_of\_starwars.sort\_values('Correlation',ascending=False).head(10)

24.we will set the threshold that if no. of person who have seen the movie is greater than 100 then only predict the movie otherwise no.

corr\_star[corr\_star['count']>100].sort\_values('Correlation',ascending=False)

25.we will make one predict function by taking movie name.

def predict\_movies(movie\_name):

#to make the matrix

movie\_user\_rating=movirate[movie\_name]

#to correlate with other movie

similar\_to\_movie=movirate.corrwith(movie\_user\_rating)

#to convert data into dataframe

convert\_corr\_of\_movie=pd.DataFrame(similar\_to\_movie,columns=['Correlation'])

#to drop na

convert\_corr\_of\_movie.dropna(inplace=True)

#to join the rating

convert\_corr\_of\_movie=convert\_corr\_of\_movie.join(rating['count'])

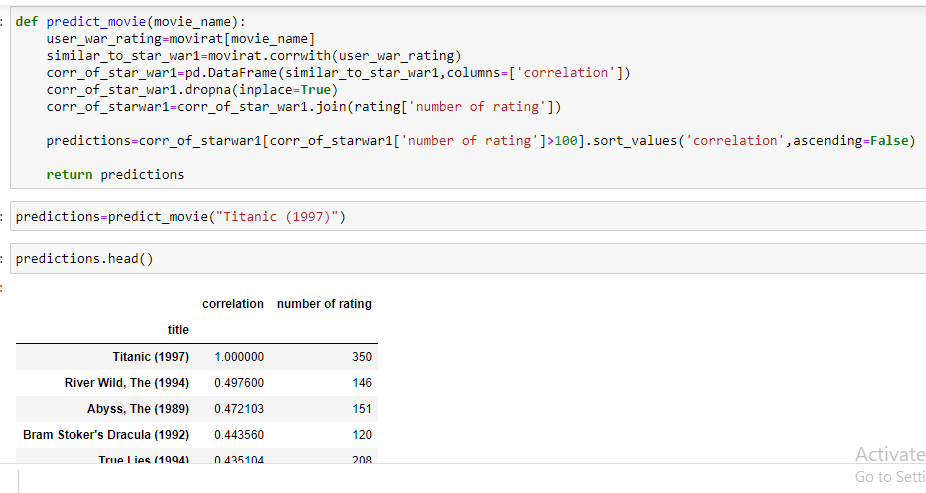
#to filter the movie seen by more than 100 user in descending order

predictions=convert\_corr\_of\_movie[convert\_corr\_of\_movie['count']>100].sort\_values('Correlation',

ascending=False)

# return prediction

return predictions



26. checking the prection

prediction=predict\_movies("Titanic (1997)")

prediction.head()