HW-3_Weekend-movie-trip

Report

Pushkar Singh Negi

Ku ID: 2946319

EECS 731: Introduction to Data Science

Notebook: Weekend_movie_trip.ipynb Purpose: Deduced Additional Information, visualization and knn clustering modelling

Input raw dataset : links.csv, movies.csv, ratings.csv, tags
Processed dataset : AdditionalInfo#1_userID_Max_rated_Movies.csv, AdditionalInfo#2_userID_Max_tagged_Movies.csv, AdditionalInfo#3_userID_max_tagged_and_max_rated.csv, AdditionalInfo#4_yearWiseUserRatingCount.csv, AdditionalInfo#5_userID_Max_rated_MoviesMonthWise.csv
Jupyter Notebook File name : Weekend_movie_trip.ipynb

1. Imported the links.csv, movies.csv, ratings.csv, tags.csv in 4 different dataframes file using pandas.
2. Checked the total non-null entries to handle missing values.
3. Handled the missing values.
4. Analyzed the dataset with the help of various commands, such as finding the total unique userId, movieId and tags.
5. Next, I found out the **Additional Information #1: To find the userId that has rated maximum number of movies**.
6. Next, I applied various pandas feature to convert the above result dataset into a panda frame.

```
In [236]: df_ratings_max_userid
```

Out[236]:

userId	Number of movies rated
414	2698
599	2478
474	2108
448	1864
274	1346
610	1302
68	1260

7. Saved the additional information data in a csv file.
8. Plotted a graph to show: number of movies rated against userId.
9. Next, I found out the **Additional Information #2: To find the userId that has tagged maximum number of movies.** with the help of groupby feature.
10. Next, I applied indexing and converted the resulted dataset into a frame.
11. After sorting the values on the basis of Number of movies tagged in the above dataset, we got to know an additional information that which userId has tagged the maximum number of movies.

Out[55]:

userId	Number of movies tagged
474	1507
567	432
62	370
599	323
477	280
424	273
537	100

12. **Visualizations:** Plotted a graph to show: number of movies tagged against userId.
13. Saved the additional information data in a csv file.
14. Next, I found out the **Additional Information #3: To find the number of movies rated and tagged by each user (df_Merged)**.
15. Applied, transformations and found an additional information: play that has maximum and minimum player

In [60]:
```
df_Merged
```

Out[60]:

userId	Numberof movies rated	Number of movies tagged
414	2698	NaN
599	2478	323.0
474	2108	1507.0
448	1864	NaN

16. **Additional Information #4: Converted the timestamp from millisecond format to 2000-07-30 18:45:03**

```
In [65]: df_ratings_timeStamp_merged_new
```

Out[65]:

	userId	movieId	rating	timestamp
0	1	1	4.0	2000-07-30 18:45:03
1	1	3	4.0	2000-07-30 18:20:47
2	1	6	4.0	2000-07-30 18:37:04
3	1	47	5.0	2000-07-30 19:03:35

17. **Created 3 new columns namely year, month and date for each of the userId.**

```
In [72]: df_ratings_timeStamp_merged_new
```

Out[72]:

	userId	movieId	rating	timestamp	year	month	date
0	1	1	4.0	2000-07-30 18:45:03	2000	7	30
1	1	3	4.0	2000-07-30 18:20:47	2000	7	30
2	1	6	4.0	2000-07-30 18:37:04	2000	7	30
3	1	47	5.0	2000-07-30 19:03:35	2000	7	30
4	1	50	5.0	2000-07-30 18:48:51	2000	7	30
5	1	70	3.0	2000-07-30 18:40:00	2000	7	30
6	1	101	5.0	2000-07-30 18:14:28	2000	7	30

18. **Additional Information #5: To find for each year how many users rated the movies (df_ratings_timeStamp_merged_newYEAR).**

```
In [82]: df_ratings_timeStamp_merged_newYEAR
```

Out[82]:

year	Number of users who rated
2000	10061
2017	8198
2007	7114
2016	6703
2015	6616
2018	6418

19. **Additional Information #6:To find for each month how many users rated the movies (df_ratings_timeStamp_merged_newMONTH).**

```
In [90]: df_ratings_timeStamp_merged_newMONTH
```

Out[90]:

month	userId
5	10883
11	9676
8	9074
3	8880
6	8825

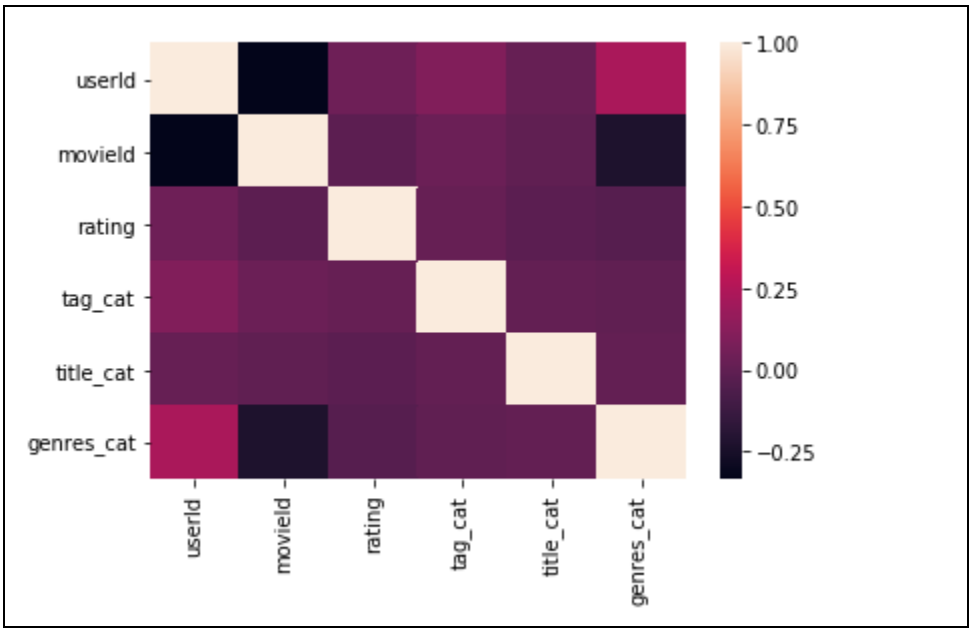
20. **Knn clustering model:** Merged the dataset and made one common dataset that contains relevant and necessary columns and discarded the less important ones.
21. In order to apply knn model, changed the datatype of all attribute to int.

22. Splited the data into testing and training data with the help of sklearn.model_selection import train_test_split

23. Fitted the X_train and y_train set

24. Found the correlation:

	userId	movieId	rating	tag_cat	title_cat	genres_cat
userId	1	-0.33	0.043	0.1	0.016	0.23
movieId	-0.33	1	-0.012	0.032	-0.0033	-0.22
rating	0.043	-0.012	1	0.02	-0.02	-0.036
tag_cat	0.1	0.032	0.02	1	0.0056	-0.00026
title_cat	0.016	-0.0033	-0.02	0.0056	1	0.0066
genres_cat	0.23	-0.22	-0.036	-0.00026	0.0066	1



25. Applied the predictions and calculated the accuracy. (0.4066 i.e. 40.66%)