ASSIGNMENT 8

RECOMMENDER SYSTEM

NAHUSHA ACHARYA PES1201700044

KEERTHAN G PES1201700963

PUSHPENDER SINGH PES1201700243

- **Collaborative filtering:** Collaborative filtering approaches build a model from user's past behavior (i.e. items purchased or searched by the user) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that user may have an interest in.
- **Content-based filtering:** Content-based filtering approaches uses a series of discrete characteristics of an item in order to recommend additional items with similar properties. Content-based filtering methods are totally based on a description of the item and a profile of the user's preferences. It recommends items based on user's past preferences.

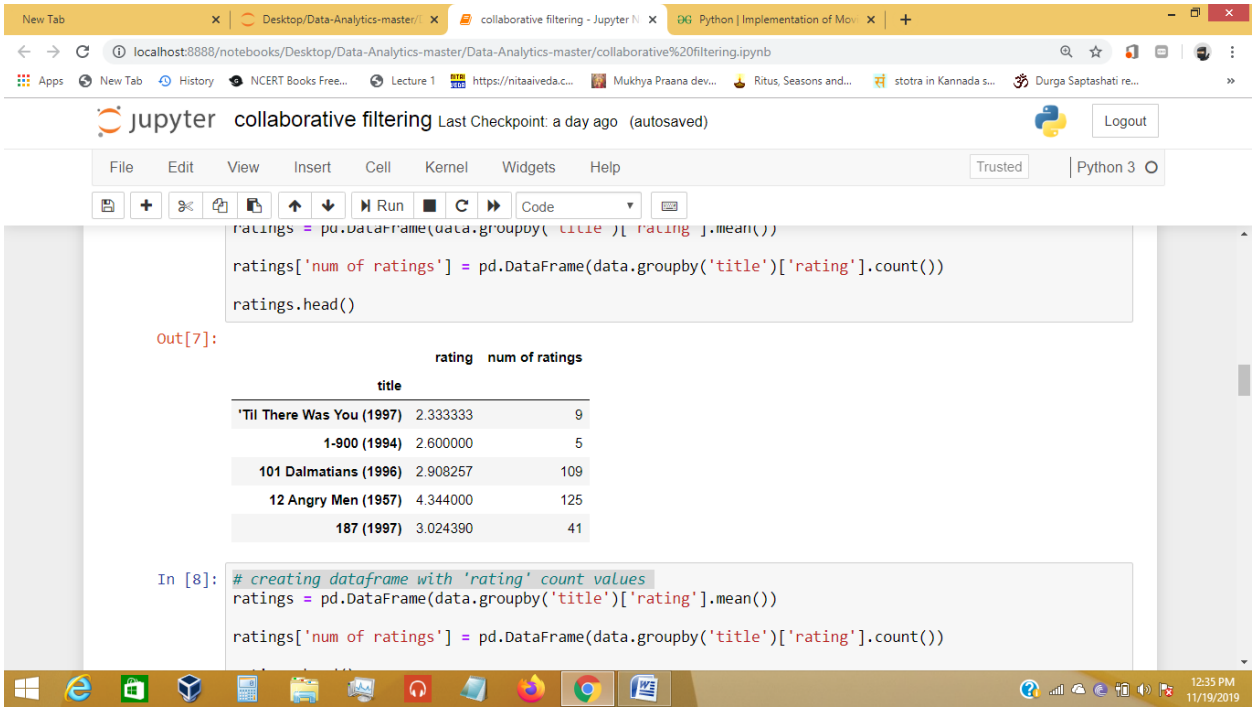
Check out all the movies and their respective IDs

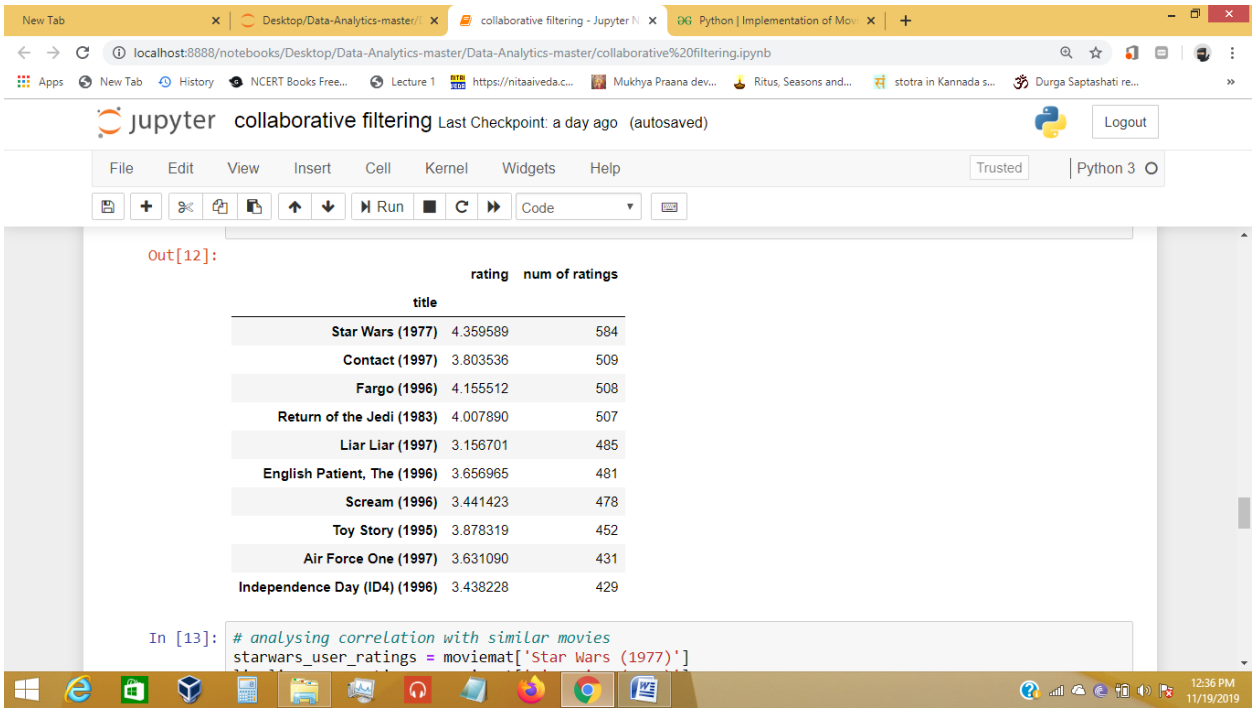
	user_id	item_id	rating	timestamp	title
0	0	50	5	881250949	Star Wars (1977)
1	290	50	5	880473582	Star Wars (1977)
2	79	50	4	891271545	Star Wars (1977)
3	2	50	5	888552084	Star Wars (1977)
4	8	50	5	879362124	Star Wars (1977)

Calculate mean rating of all movies

```
title
Marlene Dietrich: Shadow and Light (1996)     5.0
Prefontaine (1997)                            5.0
Santa with Muscles (1996)                     5.0
Star Kid (1997)                               5.0
Someone Else's America (1995)                 5.0
Name: rating, dtype: float64
```

creating dataframe with 'rating' count values





analysing correlation with similar movies

starwars_user_ratings = moviemat['Star Wars (1977)']

```
liarliar_user_ratings = moviemat['Liar Liar (1997)']
```

```
starwars_user_ratings.head()
```

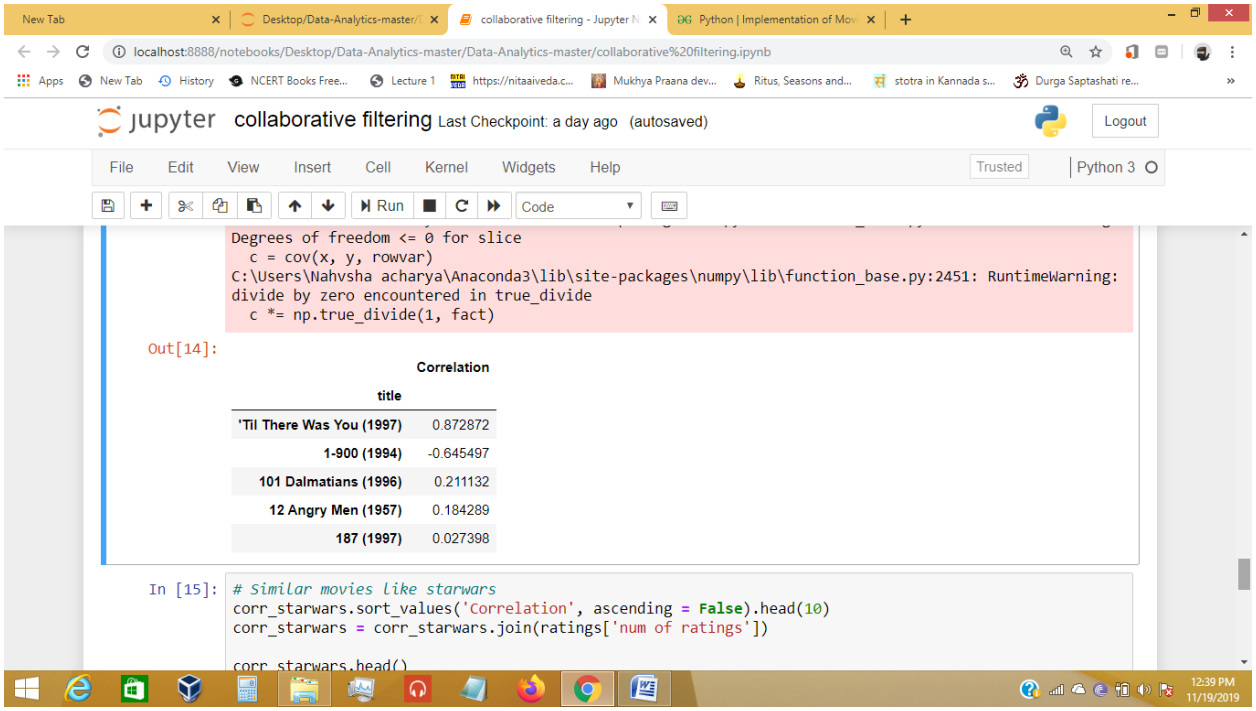
```
user_id
0    5.0
1    5.0
2    5.0
3    NaN
4    5.0
Name: Star Wars (1977), dtype: float64
```

In [14]:

```
# analysing correlation with similar movies

similar_to_starwars = moviemat.corrwith(starwars_user_ratings)

similar_to_liarliar = moviemat.corrwith(liarliar_user_ratings)
```

```
corr_starwars = pd.DataFrame(similar_to_starwars, columns =['Correlation'])

corr_starwars.dropna(inplace = True)
```

```
corr_starwars.head()
```



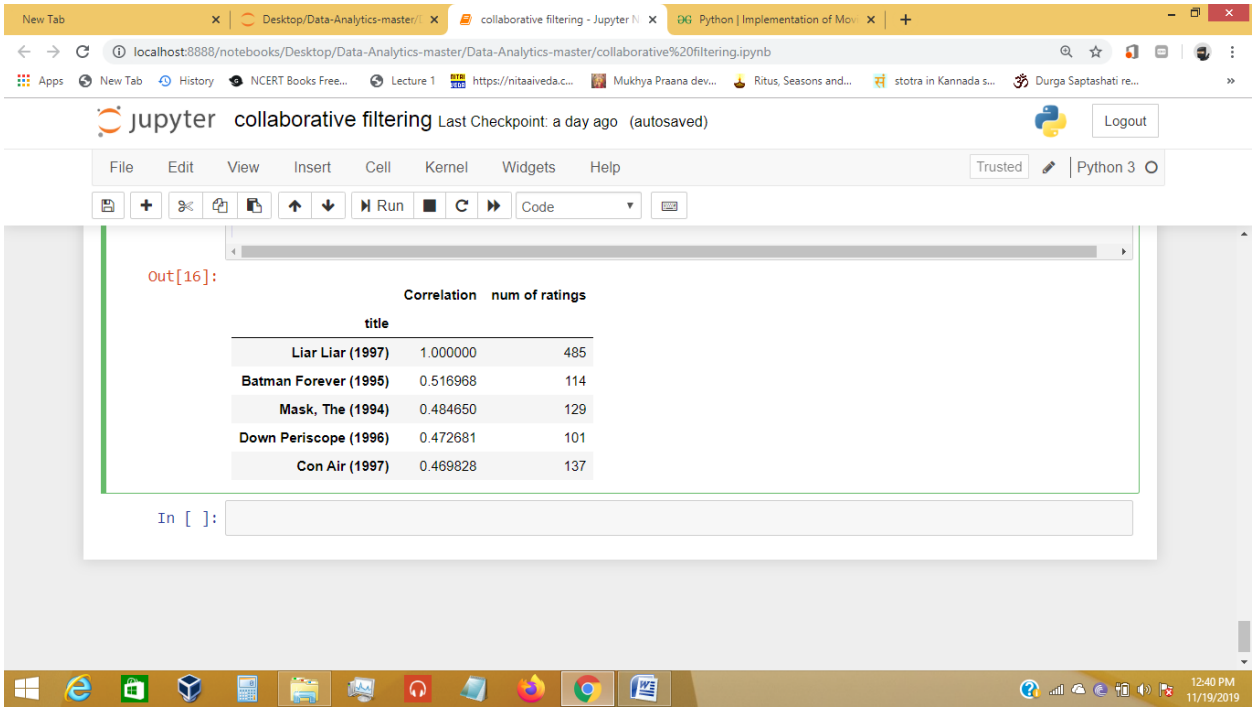
Similar movies as of liarliar

corr_liarliar = pd.DataFrame(similar_to_liarliar, columns =['Correlation'])

corr_liarliar.dropna(inplace = True)

corr_liarliar = corr_liarliar.join(ratings['num of ratings'])

corr_liarliar[corr_liarliar['num of ratings']>100].sort_values('Correlation', ascending = False).head()


New Tab
Desktop/Data-Analytics-master/
collaborative filtering - Jupyter N
Python | Implementation of Mov
localhost:8888/notebooks/Desktop/Data-Analytics-master/Data-Analytics-master/collaborative%20filtering.ipynb
Apps
New Tab
History
NCERT Books Free...
Lecture 1
https://nitaaiveda.c...
Mukhya Praana dev...
Ritus, Seasons and...
stotra in Kannada s...
Durga Saptashati re...
jupyter collaborative filtering Last Checkpoint: a day ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Out[16]:
Correlation
num of ratings
title
Liar Liar (1997) 1.000000 485
Batman Forever (1995) 0.516968 114
Mask, The (1994) 0.484650 129
Down Periscope (1996) 0.472681 101
Con Air (1997) 0.469828 137
In [ ]:
12:40 PM
11/19/2019