Exploratory Data Analysis on Movielens100k Dataset

Deep Learning Assignment - II ----> N Nikhil Kumar 18bec031(ece)

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import sklearn
```

Reading the Dataset ¶

In [2]:

```
u_cols = ['user_id', 'age', 'sex', 'occupation', 'zip_code']
r_cols = ['user_id', 'movie_id', 'rating', 'timestamp']
m_cols = ['movie_id', 'title', 'release_date', 'video_release_date', 'imdb_url']
```

In [3]:

```
users=pd.read_csv("ml-100k/u.user",encoding = "utf8",sep='|',names=u_cols)
ratings = pd.read_csv('ml-100k/u.data',encoding = "utf8",sep='\t', names=r_cols)
movies = pd.read_csv('ml-100k/u.item',encoding = 'unicode_escape', sep='|', names=m_cols, u
```

In [4]:

```
movielens=pd.merge(users,ratings)
movielens=pd.merge(movielens,movies)
movielens.head(10)
```

Out[4]:

	user_id	age	sex	occupation	zip_code	movie_id	rating	timestamp	title	release_date
0	1	24	M	technician	85711	61	4	878542420	Three Colors: White (1994)	01-Jan-1994
1	13	47	M	educator	29206	61	4	882140552	Three Colors: White (1994)	01-Jan-1994
2	18	35	F	other	37212	61	4	880130803	Three Colors: White (1994)	01-Jan-1994
3	58	27	M	programmer	52246	61	5	884305271	Three Colors: White (1994)	01-Jan-1994
4	59	49	M	educator	08403	61	4	888204597	Three Colors: White (1994)	01-Jan-1994
5	60	50	M	healthcare	06472	61	4	883326652	Three Colors: White (1994)	01-Jan-1994
6	76	20	M	student	02215	61	4	875028123	Three Colors: White (1994)	01-Jan-1994
7	94	26	M	student	71457	61	5	891720761	Three Colors: White (1994)	01-Jan-1994
8	144	53	M	programmer	20910	61	3	888106182	Three Colors: White (1994)	01-Jan-1994
9	154	25	M	student	53703	61	4	879138657	Three Colors: White (1994)	01-Jan-1994

In [5]:

```
movielens.describe()
```

Out[5]:

	user_id	age	movie_id	rating	timestamp	video_release
count	100000.00000	100000.000000	100000.000000	100000.000000	1.000000e+05	
mean	462.48475	32.969850	425.530130	3.529860	8.835289e+08	
std	266.61442	11.562623	330.798356	1.125674	5.343856e+06	
min	1.00000	7.000000	1.000000	1.000000	8.747247e+08	
25%	254.00000	24.000000	175.000000	3.000000	8.794487e+08	
50%	447.00000	30.000000	322.000000	4.000000	8.828269e+08	
75%	682.00000	40.000000	631.000000	4.000000	8.882600e+08	
max	943.00000	73.000000	1682.000000	5.000000	8.932866e+08	

In [6]:

```
movielens.shape
```

Out[6]:

```
(100000, 12)
```

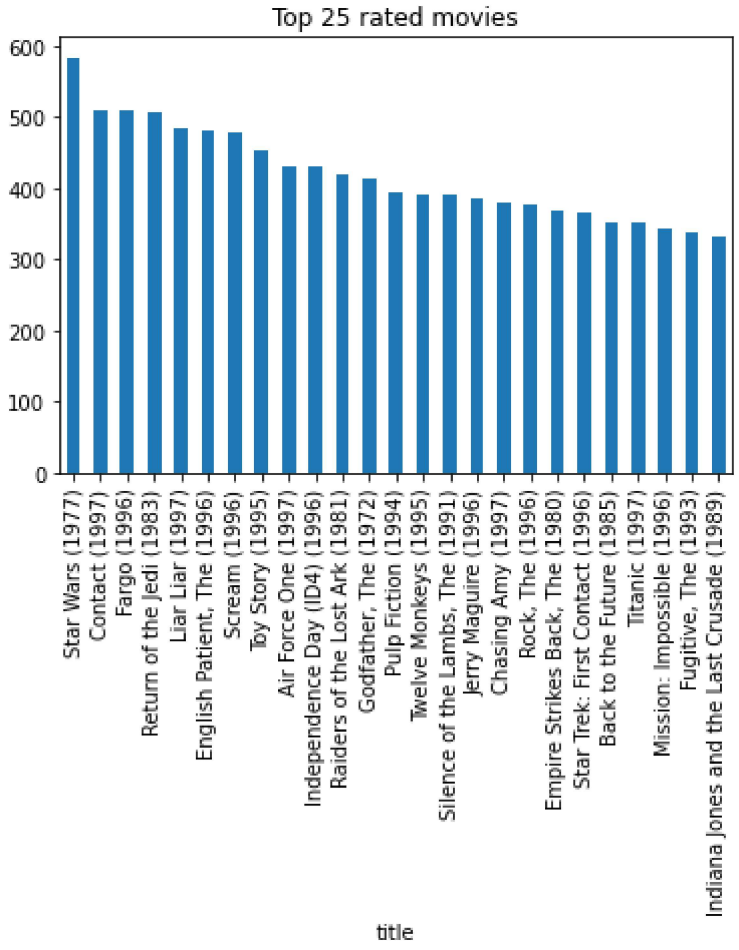
Top 25 rated movies

In [7]:

```
most_rated=movielens.groupby('title').size().sort_values(ascending=False)[:25]
most_rated.plot(kind="bar",title="Top 25 rated movies",label="count")
```

Out[7]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2366063a370>
```



In [8]:

```
movie_stat = movielens.groupby('title').agg({'rating':[np.size,np.mean]})
movie_stat.sort_values([('rating','mean')],ascending=False).head()
```

Out[8]:

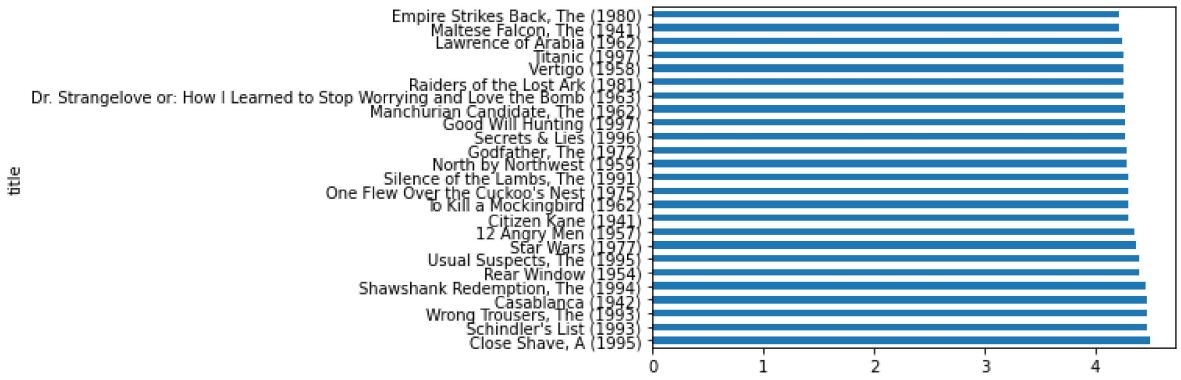
	rating	
	size	mean
title		
They Made Me a Criminal (1939)	1	5.0
Marlene Dietrich: Shadow and Light (1996)	1	5.0
Saint of Fort Washington, The (1993)	2	5.0
Someone Else's America (1995)	1	5.0
Star Kid (1997)	3	5.0

In [9]:

```
most100=movie_stat['rating']['size'] >= 100
most_rated_mean=movie_stat[most100].sort_values([('rating', 'mean')], ascending=False)
most_rated_mean['rating']['mean'].sort_values(ascending=False)[:25].plot(kind="barh")
```

Out[9]:

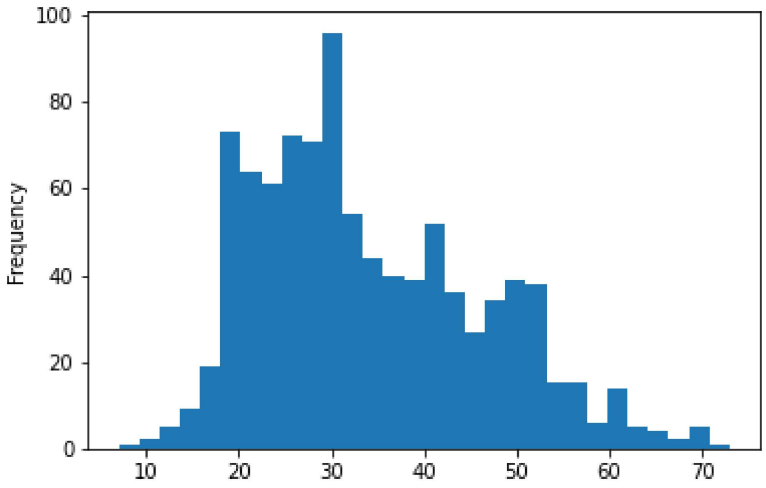
```
<matplotlib.axes._subplots.AxesSubplot at 0x2366070c040>
```



Age Distribution

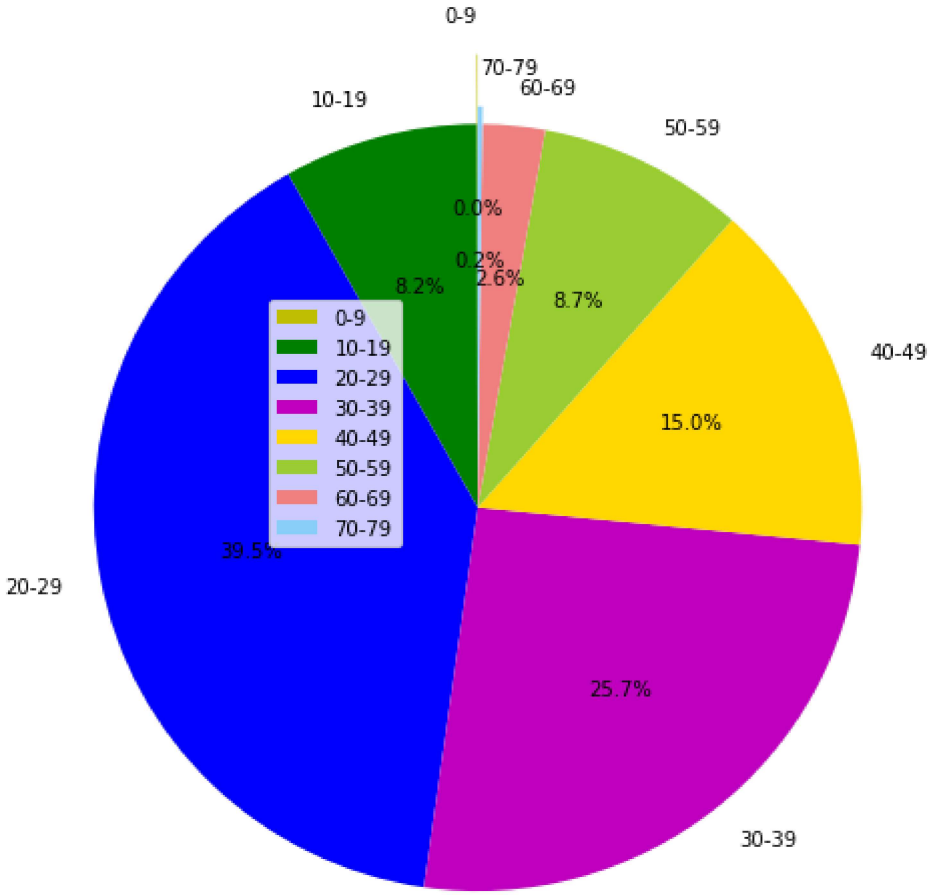
In [10]:

```
users.age.plot.hist(bins=30)
plt.xlabel=("Age")
plt.ylabel=("count")
plt.title=("Age distribution")
```



In [11]:

```
labels=['0-9','10-19','20-29','30-39','40-49','50-59','60-69','70-79']
movielens['age_group'] = pd.cut(movielens.age, range(0, 81, 10), right=False, labels=labels
distage=movielens.groupby('age_group').agg({'rating':[np.size,np.mean]})
colors=["y","g","b","m",'gold', 'yellowgreen', 'lightcoral', 'lightskyblue']
plt.pie(distage['rating']['size'],startangle=90,labels=labels,colors=colors,explode=(0.4,0,
plt.legend()
plt.show()
```



Analysis over single movie -- Toy Story

In [18]:

```
toystory=movielens[movielens.movie_id==1]
toystory.shape
mov1 = movielens.sort_values("movie_id",ascending=True).head(452) ##we have 452 reviews app
mov1.set_index('movie_id',inplace=True)
by_age = mov1.loc[mov1.index].groupby(['title', 'age_group']).agg([np.size,np.mean])
by_age
```

Out[18]:

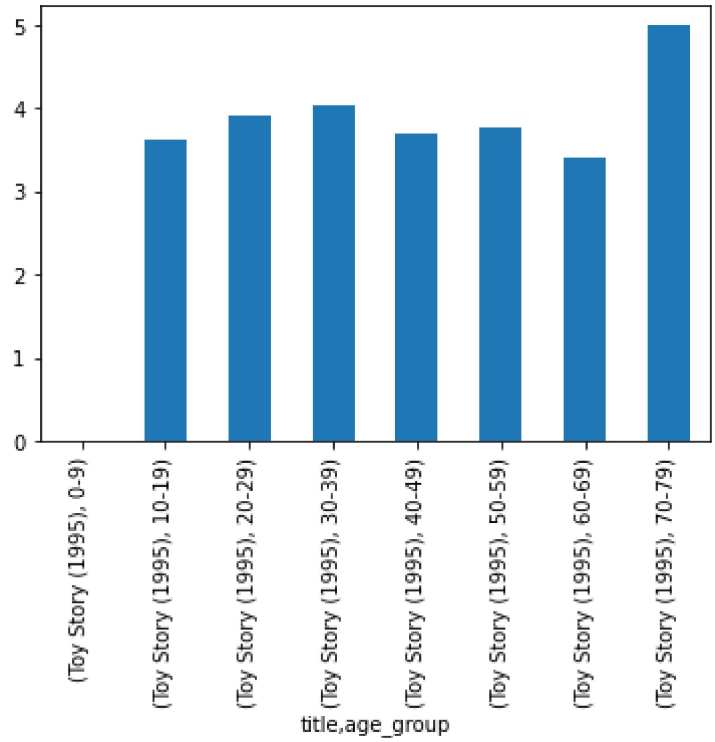
		user_id		age		rating		timestamp	
		size	mean	size	mean	size	mean	size	mean
title	age_group								
Toy Story (1995)	0-9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	10-19	16724.0	510.729730	16724.0	17.000000	16724.0	3.621622	16724.0	8.8317
	20-29	85428.0	508.132275	85428.0	24.888889	85428.0	3.920635	85428.0	8.8241
	30-39	54692.0	464.487603	54692.0	33.809917	54692.0	4.033058	54692.0	8.8219
	40-49	31640.0	433.028571	31640.0	44.528571	31640.0	3.700000	31640.0	8.8410
	50-59	13108.0	380.482759	13108.0	52.689655	13108.0	3.758621	13108.0	8.8348
	60-69	2260.0	471.800000	2260.0	61.000000	2260.0	3.400000	2260.0	8.8552
	70-79	452.0	767.000000	452.0	70.000000	452.0	5.000000	452.0	8.9146

In [19]:

```
by_age['rating']['mean'].plot(kind="bar")
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23661033460>
```



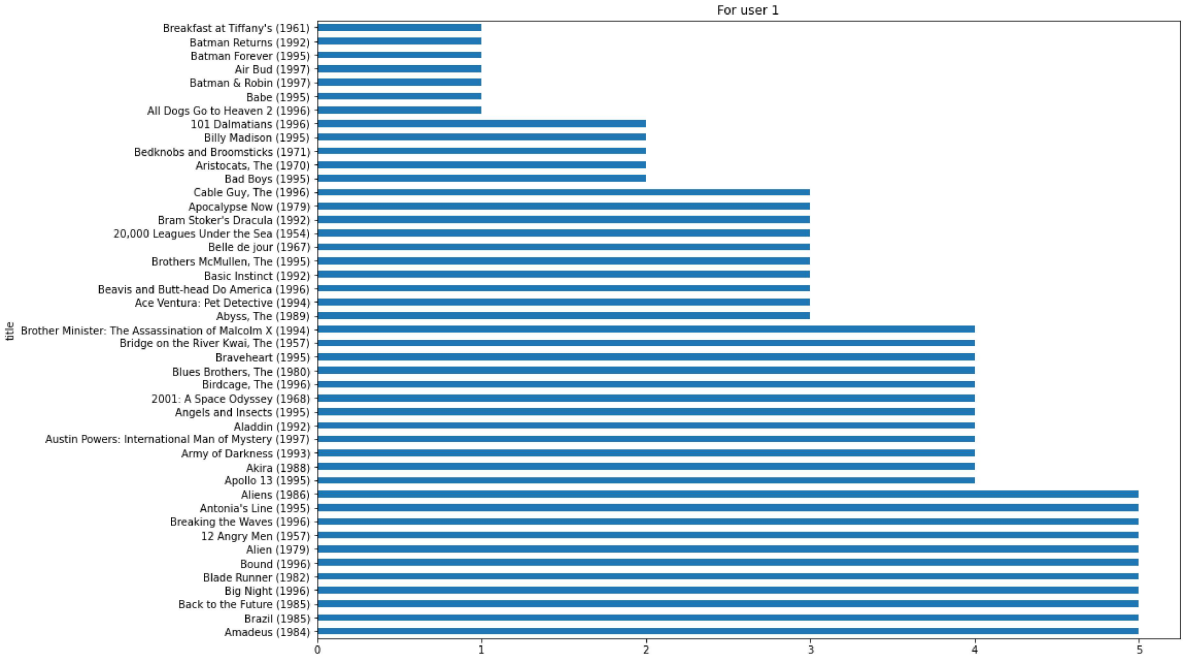
Analysis over single user -- user1

In [21]:

```
user1=movielens[movielens.user_id==1]
user1=user1.groupby('title').agg([np.size,np.mean])
user1['rating']['mean'][:45].sort_values(ascending=False).plot(kind="barh",figsize=(15,11),t
```

Out[21]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23661091df0>
```



In [ ]: