In [2]:
```
%pip install -U pip
%pip install -U setuptools wheel
%pip install pandas
%pip install glob
%pip install sklearn
```

```
Requirement already satisfied: pip in /home/studio-lab-user/.conda/envs/def
ault/lib/python3.9/site-packages (23.1.2)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: setuptools in /home/studio-lab-user/.conda/e
nvs/default/lib/python3.9/site-packages (67.8.0)
Requirement already satisfied: wheel in /home/studio-lab-user/.conda/envs/d
efault/lib/python3.9/site-packages (0.40.0)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: pandas in /home/studio-lab-user/.conda/envs/
default/lib/python3.9/site-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in /home/studio-lab-u
ser/.conda/envs/default/lib/python3.9/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /home/studio-lab-user/.cond
a/envs/default/lib/python3.9/site-packages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.20.3 in /home/studio-lab-user/.cond
a/envs/default/lib/python3.9/site-packages (from pandas) (1.23.5)
Requirement already satisfied: six>=1.5 in /home/studio-lab-user/.conda/env
s/default/lib/python3.9/site-packages (from python-dateutil>=2.8.1->pandas)
(1.12.0)
Note: you may need to restart the kernel to use updated packages.
ERROR: Could not find a version that satisfies the requirement glob (from v
ersions: none)
ERROR: No matching distribution found for glob
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: sklearn in /home/studio-lab-user/.conda/env
s/default/lib/python3.9/site-packages (0.0.post4)
Note: you may need to restart the kernel to use updated packages.
```

In [1]:
```
import pandas as pd
import glob
from sklearn.preprocessing import OneHotEncoder
import math
```

In [2]:
```
main_file_path = './DataFiles/preGenresFormat.csv'
main_df = pd.read_csv(main_file_path)
main_df.head()
```

Out[2]:

	userId	movieId	rating	title	genres	yearRatingMade	mthRa
0	1	296	5.0	Pulp Fiction (1994)	Comedy|Crime|Drama|Thriller	2006	
1	1	306	3.5	Three Colors: Red (Trois couleurs: Rouge) (1994)	Drama	2006	
2	1	307	5.0	Three Colors: Blue (Trois couleurs: Bleu) (1993)	Drama	2006	
3	1	665	5.0	Underground (1995)	Comedy|Drama|War	2006	
4	1	899	3.5	Singin' in the Rain (1952)	Comedy|Musical|Romance	2006	

In [3]:

```
test_df = main_df[:1000000]
```

In [4]:

```
def formatGenres(str):
    return str.split('|')

test_df['genresFormatted'] = test_df['genres'].apply(formatGenres)

df1 = (
    test_df['genresFormatted'].explode()
    .str.get_dummies().sum(level=0).add_prefix('Genre_')
)

test_df = test_df.drop('genresFormatted', 1).join(df1)
test_df = test_df.drop(columns='genres')
test_df.head()
```

```
/tmp/ipykernel_845/805852512.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test_df['genresFormatted'] = test_df['genres'].apply(formatGenres)
/tmp/ipykernel_845/805852512.py:7: FutureWarning: Using the level keyword i
n DataFrame and Series aggregations is deprecated and will be removed in a
future version. Use groupby instead. df.sum(level=1) should use df.groupby
(level=1).sum().
  test_df['genresFormatted'].explode()
/tmp/ipykernel_845/805852512.py:11: FutureWarning: In a future version of p
andas all arguments of DataFrame.drop except for the argument 'labels' will
be keyword-only.
  test_df = test_df.drop('genresFormatted', 1).join(df1)
```

Out[4]:

	userId	movieId	rating	title	yearRatingMade	mthRatingMade	averageRating	n
0	1	296	5.0	Pulp Fiction (1994)	2006	Apr	8.9	2
1	1	306	3.5	Three Colors: Red (Trois couleurs: Rouge) (1994)	2006	Apr	8.1	
2	1	307	5.0	Three Colors: Blue (Trois couleurs: Bleu) (1993)	2006	Apr	7.9	
3	1	665	5.0	Underground (1995)	2006	Apr	8.1	
4	1	899	3.5	Singin' in the Rain (1952)	2006	Apr	8.3	2

5 rows × 29 columns

In [5]:

```python
def formatMonths(str):
    if(str == 'Jan'):
        return 1
    if(str == 'Feb'):
        return 2
    if(str == 'Mar'):
        return 3
    if(str == 'Apr'):
        return 4
    if(str == 'May'):
        return 5
    if(str == 'Jun'):
        return 6
    if(str == 'Jul'):
        return 7
    if(str == 'Aug'):
        return 8
    if(str == 'Sep'):
        return 9
    if(str == 'Oct'):
        return 10
    if(str == 'Nov'):
        return 11
    if(str == 'Dec'):
        return 12

    return None

test_df['mthRatingMade'] = test_df['mthRatingMade'].apply(formatMonths)
```

```
test_df.head()
```

Out[5]:

	userId	movieId	rating	title	yearRatingMade	mthRatingMade	averageRating	n
0	1	296	5.0	Pulp Fiction (1994)	2006	4	8.9	2
1	1	306	3.5	Three Colors: Red (Trois couleurs: Rouge) (1994)	2006	4	8.1	
2	1	307	5.0	Three Colors: Blue (Trois couleurs: Bleu) (1993)	2006	4	7.9	
3	1	665	5.0	Underground (1995)	2006	4	8.1	
4	1	899	3.5	Singin' in the Rain (1952)	2006	4	8.3	2

5 rows × 29 columns

In [6]:

```
def userLikedTheMovie(num):
    if (num > 3.5):
        return int(1)
    else:
        return int(0)

test_df['rating'] = test_df['rating'].apply(userLikedTheMovie)

test_df.head()
```

Out[6]:

	userId	movieId	rating	title	yearRatingMade	mthRatingMade	averageRating	n
0	1	296	1	Pulp Fiction (1994)	2006	4	8.9	2
1	1	306	0	Three Colors: Red (Trois couleurs: Rouge) (1994)	2006	4	8.1	
2	1	307	1	Three Colors: Blue (Trois couleurs: Bleu) (1993)	2006	4	7.9	
3	1	665	1	Underground (1995)	2006	4	8.1	
4	1	899	0	Singin' in the Rain (1952)	2006	4	8.3	2

5 rows × 29 columns

In [8]:
```
test_df.rename(columns={'rating': 'userLikedTheMovie'}, inplace=True)
test_df.head()
```

Out[8]:

	userId	movieId	userLikedTheMovie	title	yearRatingMade	mthRatingMade	aver
0	1	296	1	Pulp Fiction (1994)	2006	4	
1	1	306	0	Three Colors: Red (Trois couleurs: Rouge) (1994)	2006	4	
2	1	307	1	Three Colors: Blue (Trois couleurs: Bleu) (1993)	2006	4	
3	1	665	1	Underground (1995)	2006	4	
4	1	899	0	Singin' in the Rain (1952)	2006	4	

5 rows × 29 columns

In [9]:
```
test_df.to_csv('./DataFiles/modelTestDataMthFormatted.csv', index=False)  #
```