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 [5]:
```
import pandas as pd
import glob
from sklearn.preprocessing import OneHotEncoder
import math
```

In [6]:
```
main_file_path = './ratings.csv'
secondary_file_path = './DataFiles/links.csv'
secondary_df = pd.read_csv(secondary_file_path)
main_df = pd.read_csv(main_file_path)
main_df = pd.merge(main_df, secondary_df, on='movieId', how='left')
main_df.head()
```

Out[6]:

	userId	movieId	rating	timestamp	imdbId	tmdbId
0	1	296	5.0	1147880044	110912	680.0
1	1	306	3.5	1147868817	111495	110.0
2	1	307	5.0	1147868828	108394	108.0
3	1	665	5.0	1147878820	114787	11902.0
4	1	899	3.5	1147868510	45152	872.0

In [7]:
```
secondary_file_path = './DataFiles/movies.csv'
secondary_df = pd.read_csv(secondary_file_path)
main_df = pd.merge(main_df, secondary_df, on='movieId', how='left')
main_df = main_df.drop(columns='tmdbId')
main_df.head()
```

Out[7]:

	userId	movieId	rating	timestamp	imdbId	title	genres
0	1	296	5.0	1147880044	110912	Pulp Fiction (1994)	Comedy|Crime|Drama|Thriller
1	1	306	3.5	1147868817	111495	Three Colors: Red (Trois couleurs: Rouge) (1994)	Drama
2	1	307	5.0	1147868828	108394	Three Colors: Blue (Trois couleurs: Bleu) (1993)	Drama
3	1	665	5.0	1147878820	114787	Underground (1995)	Comedy|Drama|War
4	1	899	3.5	1147868510	45152	Singin' in the Rain (1952)	Comedy|Musical|Romance

In [8]:
```
def getYearSubmitted(num):
    yearsSince = num/31500000
    return math.floor(yearsSince) + 1970

def getMthSubmitted(num):
    yearsSince = num/31500000
    monthDecimal = yearsSince - math.floor(yearsSince)
    return math.floor(monthDecimal*10 - 1)

main_df['yearRatingMade'] = main_df['timestamp'].apply(getYearSubmitted)
main_df['mthRatingMade'] = main_df['timestamp'].apply(getMthSubmitted)
main_df = main_df.drop(columns='timestamp')

main_df.head()
```

Out[8]:

	userId	movieId	rating	imdbId	title	genres	yearRatingMad
0	1	296	5.0	110912	Pulp Fiction (1994)	Comedy|Crime|Drama|Thriller	200
1	1	306	3.5	111495	Three Colors: Red (Trois couleurs: Rouge) (1994)	Drama	200
2	1	307	5.0	108394	Three Colors: Blue (Trois couleurs: Bleu) (1993)	Drama	200
3	1	665	5.0	114787	Underground (1995)	Comedy|Drama|War	200
4	1	899	3.5	45152	Singin' in the Rain (1952)	Comedy|Musical|Romance	200

In [9]:
```
imdb_file_path = './DataFiles/imdb.ratings.csv'
imdb_df = pd.read_csv(imdb_file_path)


def convertId(str):
    return int(str[2:])

imdb_df['imdbId'] = imdb_df['tconst'].apply(convertId)
imdb_df = imdb_df.drop(columns='tconst')
imdb_df.head()
```

Out[9]:

	averageRating	numVotes	imdbId
0	5.7	1976	1
1	5.8	264	2
2	6.5	1825	3
3	5.6	178	4
4	6.2	2617	5

In [ ]:
```
main_df = pd.merge(main_df, imdb_df, on='imdbId', how='left')
main_df = main_df.drop(columns='imdbId')
main_df.head()
```

In [ ]:
```
def getReleaseYear(str):
    yearFound = False
    inspect = str
    openIdx = None
    closeIdx = None
    releaseYear = None
    i = 5
    while(not yearFound and len(inspect) > 0):
```

```
        openIdx = inspect.find("(")
        closeIdx = inspect.find(")")
        releaseYear = inspect[openIdx+1 : closeIdx]
        i = i -1
        if (i < 0):
            return
        try:
            if (int(releaseYear) > 0):
                yearFound = True
        except:
            inspect = inspect[closeIdx + 1:]
            releaseYear = None
    return releaseYear
```

In [ ]:
```
main_df['movieReleaseYear'] = main_df['title'].apply(getReleaseYear)
main_df.head()
```

Out[ ]:

	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 [ ]:
```
main_df.to_csv('./DataFiles/preGenresFormat.csv', index=False)  # Replace 'm
```