MovieLens

Python Code to clean the Data for MovieLens

In [1]:

```
import pandas as pd
import numpy as np

df_movies = pd.read_csv('movies.csv',  encoding = 'ISO-8859-1')
df_ratings = pd.read_csv('ratings.csv',  encoding = 'ISO-8859-1')
df_users = pd.read_csv('users.csv',  encoding = 'ISO-8859-1')
```

In [2]:

```
print(df_users.shape)
print(df_users[:10])
```

```
(6040, 5)
   userID gender  age  occupation zip_code
0       1      F    1          10    48067
1       2      M   56          16    70072
2       3      M   25          15    55117
3       4      M   45           7    02460
4       5      M   25          20    55455
5       6      F   50           9    55117
6       7      M   35           1    06810
7       8      M   25          12    11413
8       9      M   25          17    61614
9      10      F   35           1    95370
```

In [3]:

```
for index, row in df_movies.iterrows():
    if index == 0:
        final_cat = row[2].split("|")
    else:
        for word in row[2].split("|"):
            if not word in final_cat:
                final_cat.append(word)

df = pd.DataFrame(columns = final_cat)
list_cat = [0]*(len(final_cat))
print(df.columns)

for index, row in df_movies.iterrows():
    df.loc[index] = list_cat
    for word in row[2].split("|"):
        if word in df.columns:
            df.loc[index][word] = 1
```

```
Index(['Animation', 'Children's', 'Comedy', 'Adventure', 'Fantasy',
'Romance',
       'Drama', 'Action', 'Crime', 'Thriller', 'Horror', 'Sci-Fi',
       'Documentary', 'War', 'Musical', 'Mystery', 'Film-Noir', 'Wes
tern'],
      dtype='object')
```

In [4]:

```
print(df[:3])
```

```
  Animation Children's Comedy Adventure Fantasy Romance Drama Action
Crime  \
0         1           1      1         0       0       0     0       0
0
1         0           1      0         1       1       0     0       0
0
2         0           0      1         0       0       1     0       0
0

  Thriller Horror Sci-Fi Documentary War Musical Mystery Film-Noir W
estern
0        0       0      0           0   0       0       0          0
0
1        0       0      0           0   0       0       0          0
0
2        0       0      0           0   0       0       0          0
0
```

In [5]:

```
import re
final_year = []
final_title = []
for index, row in df_movies.iterrows():
    word = row[1]
    if '(' in word:
        final_year.append(word[-5:-1])
        final_title.append(word[:-7])
    else:
        final_year.append("")
        final_title.append(word)
```

In [6]:

```
print(final_year[:5])
print(final_title[:5])
```

```
['1995', '1995', '1995', '1995', '1995']
['Toy Story', 'Jumanji', 'Grumpier Old Men', 'Waiting to Exhale', 'F
ather of the Bride Part II']
```

In [7]:

```
movies_cat = df_movies.loc[:,df_movies.columns[:1]]
movies_cat['year'] = final_year
movies_cat['title'] = final_title
movies = pd.concat([movies_cat, df], axis=1)
```

In [8]:

```
print(movies[:3])
print(df_users[:3])
print(df_ratings[:3])
```

```
   movieID  year             title Animation Children's Comedy Adven
ture  \
0        1  1995         Toy Story         1           1      1
0
1        2  1995           Jumanji         0           1      0
1
2        3  1995  Grumpier Old Men         0           0      1
0

  Fantasy Romance Drama   ...   Crime Thriller Horror Sci-Fi Documen
tary War  \
0       0       0     0   ...       0         0      0      0
0   0
1       1       0     0   ...       0         0      0      0
0   0
2       0       1     0   ...       0         0      0      0
0   0

  Musical Mystery Film-Noir Western
0       0       0         0       0
1       0       0         0       0
2       0       0         0       0

[3 rows x 21 columns]
   userID gender  age  occupation zip_code
0       1      F    1          10    48067
1       2      M   56          16    70072
2       3      M   25          15    55117
   userID  movieID  ratings  timestamp
0       1     1193        5  978300760
1       1      661        3  978302109
2       1      914        3  978301968
```

In [9]:

```
result = pd.merge(df_users, df_ratings, on='userID')
result = pd.merge(result, movies, on='movieID')
```

In [11]:

```
result = pd.DataFrame(result)
print("Size of the table: ", result.shape)
result.to_csv('movieLensTable.csv', index=False)
```

```
Size of the table:  (1000209, 28)
```