DSC 540- Milestone-4 API

Kannur, Gyan

Instructor Catherine Williams


```
import pandas as pd
import requests
import traceback
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

Load the previously webscraped clean data

```
web_scraped_data = pd.read_csv("./project_datasets/clean-
webscraped.csv")
```

Step 1: Reading API data using "requests" library available in Python

Before invoking the API, you need to signup for your TMDB account and get your api key to work with tmdb API.

```
api_key = '5a51d21e02c0baae3a96d6a5e7687635'
genre_dict = dict()

genre_url=f'https://api.themoviedb.org/3/genre/movie/list?
api_key={api_key}'

try:
    response = requests.get(genre_url)
    if response.status_code == 200:
        genre_dict = { genre_json['id']:genre_json['name'] for
genre_json in response.json()['genres'] }
except Exception as e:
    print('Error downloading genres: ',e)
    traceback.print_exc()

#Lets add a unique key for movies without any genre
if 0 not in genre_dict.keys():
    genre_dict[0] = 'unknown'
```

```
#Genre Names
genre_dict
```

```
{'28': 'Action',
 '12': 'Adventure',
 '16': 'Animation',
 '35': 'Comedy',
 '80': 'Crime',
 '99': 'Documentary',
 '18': 'Drama',
 '10751': 'Family',
 '14': 'Fantasy',
 '36': 'History',
 '27': 'Horror',
 '10402': 'Music',
 '9648': 'Mystery',
 '10749': 'Romance',
 '878': 'Science Fiction',
 '10770': 'TV Movie',
 '53': 'Thriller',
 '10752': 'War',
 '37': 'Western',
 '0': 'unknown'}
```

```
#Display the standard list of genres

genre_dict
```

```
{'28': 'Action',
 '12': 'Adventure',
 '16': 'Animation',
 '35': 'Comedy',
 '80': 'Crime',
 '99': 'Documentary',
 '18': 'Drama',
 '10751': 'Family',
 '14': 'Fantasy',
 '36': 'History',
 '27': 'Horror',
 '10402': 'Music',
 '9648': 'Mystery',
 '10749': 'Romance',
 '878': 'Science Fiction',
 '10770': 'TV Movie',
 '53': 'Thriller',
 '10752': 'War',
 '37': 'Western',
 '0': 'unknown'}
```

```
#Get the list of unique movie titles
scraped_movie_list=web_scraped_data['movie_title'].unique()
len(scraped_movie_list)
```

```
353
```

Fetch the movie details using the api keys

```
%%time
# We will first create an empty dataframe to store all the movie
detail
api_df = pd.DataFrame()
# Our for loop will iterate through each page, get json data convert
it into dataframe and append it to original dataframe
for title in scraped_movie_list:
    url = f"https://api.themoviedb.org/3/search/movie?
api_key={api_key}&query={title}"
    response = requests.get(url)
    if 'results' in response.json():
        temporary_df = pd.DataFrame(response.json()['results'])
        api_df = pd.concat([api_df,temporary_df],ignore_index=True)
```

```
CPU times: total: 984 ms
Wall time: 34.1 s
```

```
api_df.head()
```

```
   adult                         backdrop_path  \
0  False  /npCPnwDyWfQltGfIZKN6WqeUXGI.jpg
1  False  /u2bZhH3nTf0So0UIC1QxAqBvC07.jpg
2  False  /vD1yKObsRS2cvpmtuaCaMhr4zxe.jpg
3  False  /sGjhRHNiQSkgVec18D3oX45hPmz.jpg
4  False                                None

                                  genre_ids      id
original_language  \
0             [Fantasy, Adventure, Action]   57158                  en

1  [Animation, Family, Adventure, Fantasy]  109445                  en

2                                [Thriller]   44363                  en

3                                [Thriller]   26041                  en

4                           [Drama, Horror]  170986                  hi


                         original_title  \
0  The Hobbit: The Desolation of Smaug
1                               Frozen
2                               Frozen
```

```
3                                Frozen
4                                Frozen

                                                overview  popularity  \
0  The Dwarves, Bilbo and Gandalf have successful...      87.340
1  Young princess Anna of Arendelle dreams about ...     191.622
2  When three skiers find themselves stranded on ...      40.401
3  It's two years since the mysterious disappeara...       7.794
4  This is a touching and somber journey of Lasya...       4.000

                         poster_path release_date  \
0  /xQYiXsheRCDBA39DOrmaw1aSpbk.jpg   2013-12-11
1  /mmWheq3cFI4tYrZDiATOkCNTqgK.jpg   2013-11-20
2  /2J3URUnDrIpNvh0uVqINQvr4HhW.jpg   2010-02-05
3  /a6RlPQUerliQLkAieku5B8Loamk.jpg   2005-03-12
4  /2GL9yZtrgbYKCeKBc3TF9gGfZpX.jpg   2007-07-21

                                 title  video  vote_average
vote_count
0  The Hobbit: The Desolation of Smaug  False         7.574
13227.0
1                                Frozen  False         7.246
16611.0
2                                Frozen  False         5.996
1831.0
3                                Frozen  False         5.700
18.0
4                                Frozen  False         7.200
4.0

api_df.shape

(3001, 14)
```

Check if the genre_ids have any empty data like [] as genre_ids column is a list

```
api_df.genre_ids.value_counts()

genre_ids
[unknown]                                                    329
[Documentary]                                                262
[Drama]                                                      190
[Comedy]                                                     128
[Horror]                                                     125
                                                            ...
[Documentary, History, Crime]                                  1
[Crime, Thriller, Action]                                      1
[TV Movie, Action, Adventure, Fantasy, Science Fiction]        1
[Action, Animation, Fantasy, Science Fiction]                  1
[Animation, Fantasy, Comedy]                                   1
Name: count, Length: 761, dtype: int64
```

pandas replace empty square brackets with [0], here genre[0] is unknown which we created earlier

```
# pandas replace empty square brackets with [0], here genre[0] is
unknown which we created earlier
api_df['genre_ids'] = api_df['genre_ids'].apply(lambda  x : [0] if not
x else x)

api_df.genre_ids.value_counts()
```

```
genre_ids
[unknown]                                                          329
[Documentary]                                                      262
[Drama]                                                            190
[Comedy]                                                           128
[Horror]                                                           125
                                                                  ...
[Documentary, History, Crime]                                        1
[Crime, Thriller, Action]                                            1
[TV Movie, Action, Adventure, Fantasy, Science Fiction]              1
[Action, Animation, Fantasy, Science Fiction]                        1
[Animation, Fantasy, Comedy]                                         1
Name: count, Length: 761, dtype: int64
```

Removing brackets [] from list type inside pandas cell

```
#removing brackets [] from list type inside pandas cell
api_df['genre_ids'] = api_df['genre_ids'].str.join(',')

#List out the genre and its count, observe there are no square braces
[]
api_df['genre_ids'].value_counts()
```

```
genre_ids
unknown                                                    329
Documentary                                                262
Drama                                                      190
Comedy                                                     128
Horror                                                     125
                                                         ...
Documentary,History,Crime                                    1
Crime,Thriller,Action                                        1
TV Movie,Action,Adventure,Fantasy,Science Fiction            1
Action,Animation,Fantasy,Science Fiction                     1
Animation,Fantasy,Comedy                                     1
Name: count, Length: 761, dtype: int64
```

Extract year as a separate column from release_date

```
#extract year as a separate column from release_date
api_df['release_date']=pd.to_datetime(api_df['release_date'],format="%
Y-%m-%d")
```

```
api_df["release_year"] = pd.to_datetime(api_df['release_date'],
format="%Y-%m-%d").dt.year
api_df.head()
```

```
   adult                     backdrop_path  \
0  False  /npCPnwDyWfQltGfIZKN6WqeUXGI.jpg
1  False  /u2bZhH3nTf0So0UIC1QxAqBvC07.jpg
2  False  /vD1yKObsRS2cvpmtuaCaMhr4zxe.jpg
3  False  /sGjhRHNiQSkgVec18D3oX45hPmz.jpg
4  False                               None

                          genre_ids      id original_language  \
0           Fantasy,Adventure,Action   57158                en
1  Animation,Family,Adventure,Fantasy  109445                en
2                           Thriller   44363                en
3                           Thriller   26041                en
4                      Drama,Horror  170986                hi

                      original_title  \
0  The Hobbit: The Desolation of Smaug
1                              Frozen
2                              Frozen
3                              Frozen
4                              Frozen

                                            overview  popularity  \
0  The Dwarves, Bilbo and Gandalf have successful...      87.340
1  Young princess Anna of Arendelle dreams about ...     191.622
2  When three skiers find themselves stranded on ...      40.401
3  It's two years since the mysterious disappeara...       7.794
4  This is a touching and somber journey of Lasya...       4.000

                      poster_path release_date  \
0  /xQYiXsheRCDBA39DOrmaw1aSpbk.jpg   2013-12-11
1  /mmWheq3cFI4tYrZDiATOkCNTqgK.jpg   2013-11-20
2  /2J3URUnDrIpNvh0uVqINQvr4HhW.jpg   2010-02-05
3  /a6RlPQUerliQLkAieku5B8Loamk.jpg   2005-03-12
4  /2GL9yZtrgbYKCeKBc3TF9gGfZpX.jpg   2007-07-21

                               title  video  vote_average
vote_count  \
0  The Hobbit: The Desolation of Smaug  False         7.574
13227.0
1                              Frozen  False         7.246
16611.0
2                              Frozen  False         5.996
1831.0
3                              Frozen  False         5.700
18.0
4                              Frozen  False         7.200
```

```
4.0

   release_year
0        2013.0
1        2013.0
2        2010.0
3        2005.0
4        2007.0

#Check if there are any nan in release year
api_df.loc[api_df.release_year.isna()]

      adult                       backdrop_path  \
5     False                                None
32    False                                None
75    False  /xcIqtToUFrie1o4g4ZtYVKj5R1f.jpg
88    False                                None
113   False                                None
...     ...                                 ...
2902  False  /46Br5afTklXva5gRI6wXcAApaGP.jpg
2924  False                                None
2925  False                                None
2974  False                                None
2976  False                                None

                                       genre_ids       id
original_language  \
5                                        unknown   950554
en
32                                       unknown   566990
en
75                         Science Fiction,Action   374771
en
88                                 Comedy,Romance   887567
en
113                                      unknown  1159811
ko
...                                          ...      ...
...
2902  Animation,Action,Adventure,Science Fiction   911916
en
2924            Action,Adventure,Science Fiction   939345
en
2925    Action,Adventure,Science Fiction,Fantasy   939347
en
2974                                     unknown  1195165
en
2976                               Comedy,Horror   601823
en
```

```
                           original_title  \
5                                  Frozen
32                                 Gravity
75                           Riddick: Furya
88                               The Butler
113
...                                     ...
2902  Spider-Man: Beyond the Spider-Verse
2924   Transformers: Rise of the Beasts 2
2925   Transformers: Rise of the Beasts 3
2974                   Dungeons & Derrick
2976                Playdate in a Dungeon

                                                 overview  popularity  \
5                            A film by Adonia Bouchehri.       0.402
32    Three boys power play with a gun. Gravity is a...       0.001
75    Riddick finally returns to his home world, a p...      16.738
88    A story of a butler who is in love with their ...       0.001
113   Srey Na, a female immigrant from Cambodia, who...       0.001
...                                                  ...         ...
2902  The third installment in the Spider-Verse fran...      29.473
2924  The first of two planned sequels to the 2023 f...      19.756
2925  The second of two planned sequels to the 2023 ...      14.596
2974  Derrick and Tori, best friends and avid player...       0.001
2976  Three strangers wake up in the basement of the...       0.600

                           poster_path release_date  \
5                                 None          NaT
32                                None          NaT
75                                None          NaT
88                                None          NaT
113                               None          NaT
...                                ...          ...
2902  /rZ4arzyaDyI8l9Y7VIPPsDGARwh.jpg          NaT
2924  /f4PFiwOHVcNUXRcOmxX2hUYdAx7.jpg          NaT
2925  /zjDGpjRj9M9pLqVVZPpaFhG6BLx.jpg          NaT
2974                              None          NaT
2976  /xT9qSpG5W1zMUEKnJsrGp7yuIlX.jpg          NaT

                                    title  video  vote_average
vote_count  \
5                                  Frozen  False           0.0
0.0
32                                 Gravity  False           0.0
0.0
75                           Riddick: Furya  False           0.0
0.0
88                               The Butler  False           0.0
0.0
113                  The Road to Elysium  False           0.0
```

```
0.0
...                                     ...    ...            ...
...
2902  Spider-Man: Beyond the Spider-Verse  False            0.0
0.0
2924   Transformers: Rise of the Beasts 2  False            0.0
0.0
2925   Transformers: Rise of the Beasts 3  False            0.0
0.0
2974                   Dungeons & Derrick  False            0.0
0.0
2976                Playdate in a Dungeon  False            0.0
0.0

      release_year
5              NaN
32             NaN
75             NaN
88             NaN
113            NaN
...            ...
2902           NaN
2924           NaN
2925           NaN
2974           NaN
2976           NaN

[150 rows x 15 columns]
```

```
#Fill in those nan with some arbitrary value so you can avoid any type conversion errors
api_df['release_year'] = api_df.release_year.fillna(1900)

#conver floats to ints now
api_df["release_year"] = api_df.release_year.astype('int64')

# checks if any of columns in the data have null values - should print False
api_df.isnull().sum().any()
```

```
True
```

```
api_df.dropna(inplace=True)

api_df.shape
```

```
(1893, 15)
```

```
api_df.head()
```

```
   adult                       backdrop_path  \
0  False  /npCPnwDyWfQltGfIZKN6WqeUXGI.jpg
1  False  /u2bZhH3nTf0So0UIC1QxAqBvC07.jpg
2  False  /vD1yKObsRS2cvpmtuaCaMhr4zxe.jpg
3  False  /sGjhRHNiQSkgVec18D3oX45hPmz.jpg
6  False  /9PxXSAnbVfvFacsGTJu1aXEWVg7.jpg

                            genre_ids      id original_language  \
0           Fantasy,Adventure,Action   57158                en
1  Animation,Family,Adventure,Fantasy  109445                en
2                           Thriller   44363                en
3                           Thriller   26041                en
6   Animation,Adventure,Comedy,Family  573171                es

                        original_title  \
0  The Hobbit: The Desolation of Smaug
1                               Frozen
2                               Frozen
3                               Frozen
6                 Huevitos Congelados

                                            overview  popularity  \
0  The Dwarves, Bilbo and Gandalf have successful...      87.340
1  Young princess Anna of Arendelle dreams about ...     191.622
2  When three skiers find themselves stranded on ...      40.401
3  It's two years since the mysterious disappeara...       7.794
6  In the final Huevos adventure, Toto and his fa...      37.370

                        poster_path release_date  \
0  /xQYiXsheRCDBA39DOrmaw1aSpbk.jpg   2013-12-11
1  /mmWheq3cFI4tYrZDiATOkCNTqgK.jpg   2013-11-20
2  /2J3URUnDrIpNvh0uVqINQvr4HhW.jpg   2010-02-05
3  /a6RlPQUerliQLkAieku5B8Loamk.jpg   2005-03-12
6  /8xCO3IarklLD4tK1rPn0e4gSMoV.jpg   2022-12-14

                                 title  video  vote_average
vote_count  \
0  The Hobbit: The Desolation of Smaug  False         7.574
13227.0
1                               Frozen  False         7.246
16611.0
2                               Frozen  False         5.996
1831.0
3                               Frozen  False         5.700
18.0
6         Little Eggs: A Frozen Rescue  False         7.670
348.0

   release_year
0          2013
```

```
1          2013
2          2010
3          2005
6          2022

api_df.columns

Index(['adult', 'backdrop_path', 'genre_ids', 'id',
'original_language',
       'original_title', 'overview', 'popularity', 'poster_path',
       'release_date', 'title', 'video', 'vote_average', 'vote_count',
       'release_year'],
      dtype='object')
```

Convert key column title to lowercase

```
api_df['title'] = api_df.title.str.lower()
```

Drop unused columns

```
api_df.drop(['id','adult','backdrop_path','poster_path','video'],
axis=1, inplace=True)

api_df.rename(columns={'genre_ids': 'genres'}, inplace=True)

new_col_order=['title', 'release_year','genres', 'popularity',
'vote_average','vote_count', 'original_title', 'overview'
                ,'release_date','original_language'
               ]

for i,col in enumerate(new_col_order):
    tmp = api_df[col]
    api_df.drop(labels=[col],axis=1,inplace=True)
    api_df.insert(i,col,tmp)

sum(api_df.duplicated())
api_df.drop_duplicates(inplace=True)

#Final data of cleaned and transformed data
api_df.head()
```

```
                                  title  release_year  \
0  the hobbit: the desolation of smaug          2013
1                                frozen          2013
2                                frozen          2010
3                                frozen          2005
6         little eggs: a frozen rescue          2022

                                genres  popularity  vote_average
vote_count  \
0           Fantasy,Adventure,Action      87.340         7.574
```

```
13227.0
1  Animation,Family,Adventure,Fantasy     191.622          7.246
16611.0
2                            Thriller      40.401          5.996
1831.0
3                            Thriller       7.794          5.700
18.0
6   Animation,Adventure,Comedy,Family      37.370          7.670
348.0

                         original_title  \
0  The Hobbit: The Desolation of Smaug
1                                Frozen
2                                Frozen
3                                Frozen
6                  Huevitos Congelados

                                              overview release_date  \
0  The Dwarves, Bilbo and Gandalf have successful...   2013-12-11
1  Young princess Anna of Arendelle dreams about ...   2013-11-20
2  When three skiers find themselves stranded on ...   2010-02-05
3  It's two years since the mysterious disappeara...   2005-03-12
6  In the final Huevos adventure, Toto and his fa...   2022-12-14

  original_language
0                en
1                en
2                en
3                en
6                es
```

Store the transformed api data to be used in future

```
api_df.to_csv(r'./project_datasets/clean-api_data.csv',index=False)
```

API Ethical Implications and Assumptions:

What changes were made to the data?

The api may provide some of the dollar amounts separate by commas. I had to create columns which tells within what range the production cost was, this user friendly representation is easier to read. Example:

production cost= 365,000,000 prod_cost_range_million = 361-366

The genres are listed as numbers, Example genre_id 28 which maps to Action and 12 maps to Adventure.

I had to call another api to fetch the standard genre list and covert the genre_ids to its equivalent names.

I also created a separate year column from the release date. The year the movie released is significant for my future analysis and visualizations.

Are there any legal or regulatory guidelines for your data or project topic?

There is a huge number of movies release every year and this would be a problem when making several calls and the api can be rate limited, hence I had to cut down the analysis to last 10 years.

What risks could be created based on the transformations done?

Care should be taken when saving the files into csv and later persisting them in database. Some of the database like SQLite does not allow a list of values to be stored.

Did you make any assumptions in cleaning/transforming the data?

I had to drop some columns such as adult - which was to indicate the age recommendations for that movie. I also dropped columns like backdrop_path and poster_path which as not part of the analysis. The genre was tricky as some movies fall under more than one. As it had square braces, that would throw an exception when I have to save it in database.

How was your data sourced / verified for credibility?

I initially signed up for an account in imdb for an api key.The api had limitations on the number of requests I can make, therefore I signed up with tmdb api key which was far more flexible. TMDB is a very well know movie analytics website catering the needs of several projects wanting the movie data.

Was your data acquired in an ethical way?

The api keys ensure the api is accessed only via verified sources. The service hosted can only be accessed via the api keys