SUBMISSION

Ingest a Dataset

1. Introduction

Step 1: Import libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
```

Step 2: Import data

The dataset I choose is **TMDb movie data**

```
data = pd.read_csv('tmdb-movies.csv')
```

2. Data Wrangling

Step 1: Write out few rows

```
data.head()
```

Step 2: Check data type

```
data.dtypes
```

Step 3: Change data type of column “release date” to date time

```
data['release_date'] = pd.to_datetime(data['release_date'])
```

Step 4: Check number of null value of each column

```
for col in data:
  num_of_null = data[col].isnull().sum()
  print(col, num_of_null)
```

As the result, we can see there is no null value at important column

Step 5: Drop some column which we don’t need

```
data.drop('id', axis=1, inplace=True)
data.drop('imdb_id', axis=1, inplace=True)
data.drop('homepage', axis=1, inplace=True)
```

3. Exploratory Data Analysis

Question 1: How is the development of cinema industry by time?

Q1.1 Problem

One of my hobbies is watching movie and I wonder: “How is the development of cinema industry by time?”

I want to see the number of films by genres and what does audience think about genres by rating and revenue,…

Q1.2 Solve

Step 1: Check all distinct value of genres column

```
genres_list = []
for index, record in data.iterrows():
  genres_of_film = str(record['genres']).split('|')
  for genres in genres_of_film:
    if genres not in genres_list:
      genres_list.append(genres)
genres_list.remove('nan')
print(genres_list)
```

Step 2: Check all distinct value of release_year column

```
years_list = []
for index, record in data.iterrows():
  year = int(record['release_year'])
  if year not in years_list:
    years_list.append(year)
years_list.sort()
# years_list = years_list.sort()
print(years_list)
```

Step 3: Because there are too many types of genres, I will combine them that base on my experience and some reference on internet

```
types = {
  'Drama': ['Drama', 'Family', 'Comedy', 'Romance', 'Music'],
  'Action/Adventure': ['Action', 'Crime', 'Western', 'Adventure', 'Science
Fiction', 'Fantasy', 'Thriller'],
  'Documentary': ['War', 'History', 'Documentary', 'Foreign'],
  'Animation': ['Animation'],
  'TV Movie': ['TV Movie'],
  'Horror': ['Horror', 'Mystery', 'Thriller']
}
```

Step 4: Calculate some metrics such as number of film, return on equity (ROE), rating, number of high rating film,…

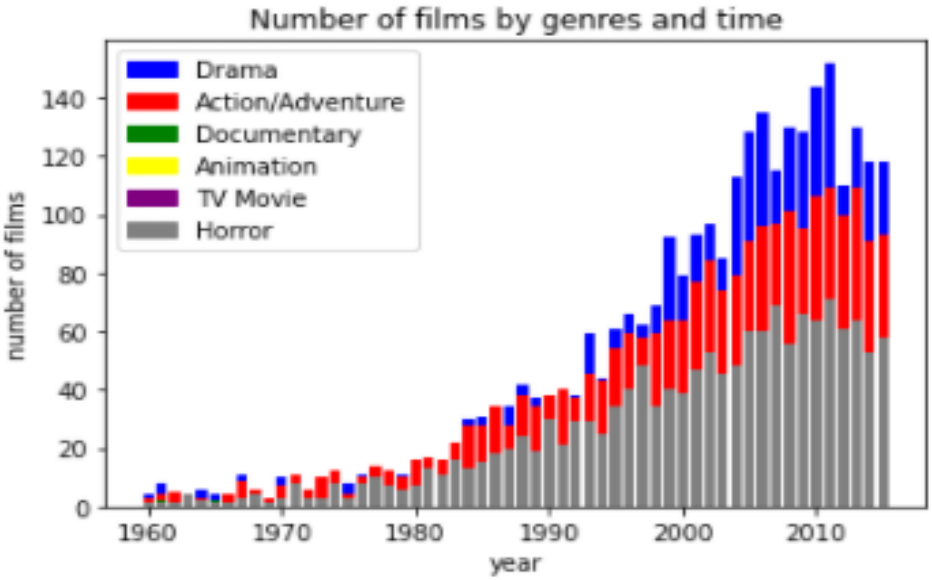
```
def generate_types_film(genres):
    genres_list = genres.split('|')
    result = []
    for ele in genres_list:
        for type in types.keys():
            if ele in types[type] and type not in result:
                result.append(type)
    return result

metrics = {}

for index, record in data.iterrows():
    year = record['release_year']
    genres = record['genres']
    rating = record['vote_average']
    highrating = 1 if rating >= 7 else 0
    type_film = generate_types_film(str(genres))
    # print(record['original_title'], type_film)
    if record['revenue'] == 0 or record['budget'] == 0:
        continue
    roe = min(record['revenue']/record['budget'] - 1, 200)
    for type in type_film:
        key = str(year) + " " + type
        if key not in metrics:
            metrics[key] = [1, roe, rating, highrating]
        else:
            metrics[key][0] += 1
            metrics[key][1] += roe
            metrics[key][2] += rating
            metrics[key][3] += highrating
```

Step 5: Create q1 dataframe

```
col_year = []
col_type = []
for year in years_list:
    for type in types.keys():
        col_year.append(year)
        col_type.append(type)

q1 = pd.DataFrame({'year': col_year, 'type': col_type})

col_count = []
col_roe = []
col_rating = []
col_highrating = []

for index, record in q1.iterrows():
    year = record['year']
    type = record['type']
    key = str(year) + " " + type
    try:
        count = metrics[key][0]
        col_count.append(count)
        roe = metrics[key][1]
        col_roe.append(round(roe/count,1))
        rating = metrics[key][2]
        col_rating.append(round(rating/count,1))
        highrating = metrics[key][3]
        col_highrating.append(round(highrating/count*100))
    except:
        col_count.append(0)
        col_roe.append(0)
        col_rating.append(0)
        col_highrating.append(0)

q1['count'] = col_count
q1['roe'] = col_roe
q1['rating'] = col_rating
q1['highrating'] = col_highrating
```

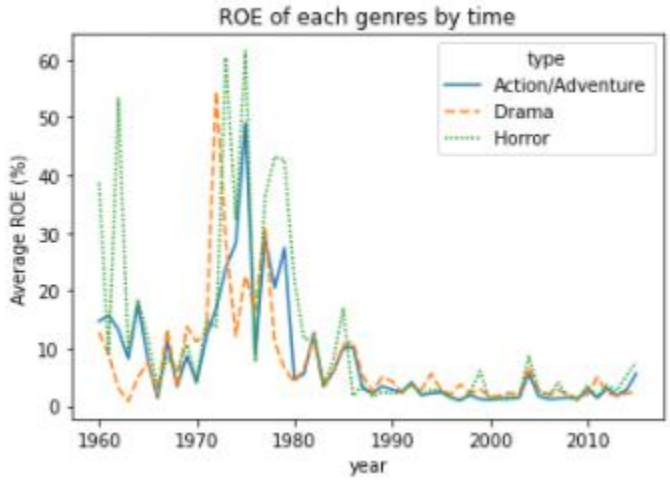
Step 6: Vizualize the “number of films by time and genres” chart

```
colors = {
  'Drama': 'blue',
  'Action/Adventure': 'red',
  # 'Adventure': 'orange',
  'Documentary': 'green',
  'Animation': 'yellow',
  'TV Movie': 'purple',
  'Horror': 'grey'
}
labels = list(colors.keys())
handles = [plt.Rectangle((0,0),1,1, color=colors[label]) for label in labels]
c = q1['type'].apply(lambda x: colors[x])

bars = plt.bar(q1['year'], q1['count'], color=c, label=labels)
plt.legend(handles, labels)
plt.title('Number of films by genres and time')
```



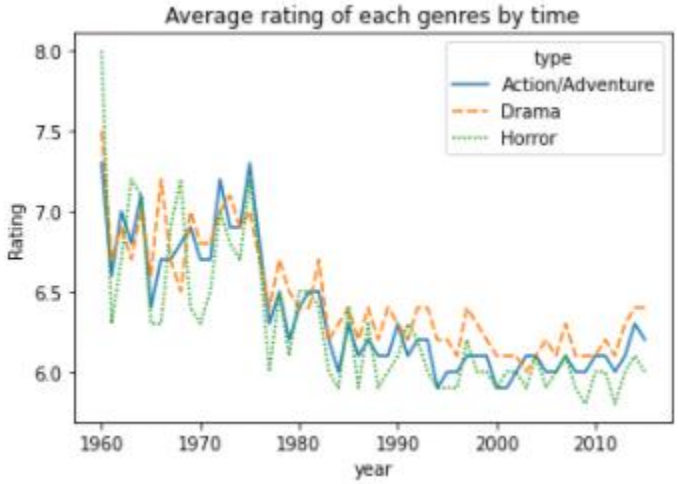
Step 7: Vizualize the “Average ROE of each genres by time“ chart

```
option = ['Horror', 'Drama', 'Action/Adventure']
q1_type = q1[q1['type'].isin(option)]
q1_roe = q1_type.pivot("year", "type", "roe")
sns.lineplot(data=q1_roe)
plt.title('Average ROE of each genres by time')
plt.xlabel('year')
plt.ylabel('Average ROE (%)')
```



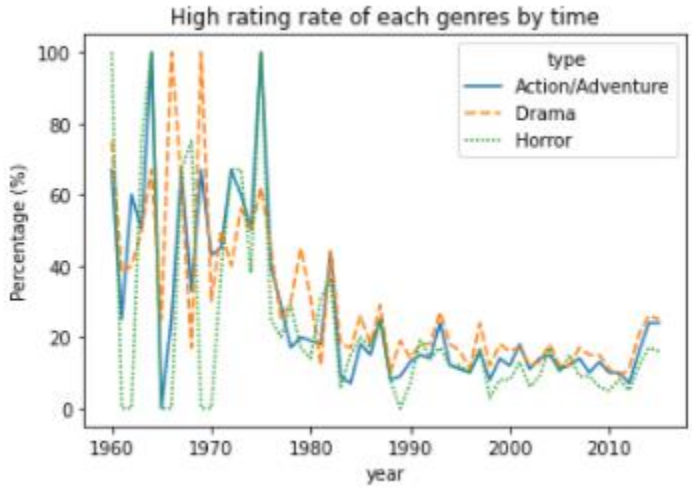
Step 8: Vizualize the “Average rating of each genres by time“ chart

```
q1_rating = q1_type.pivot("year", "type", "rating")
sns.lineplot(data=q1_rating)
plt.title('Average rating of each genres by time')
plt.xlabel('year')
plt.ylabel('Rating')
```



Step 9: Vizualize the “number of high rating films by time and genres” chart

```
q1_highrating = q1_type.pivot("year", "type", "highrating")
sns.lineplot(data=q1_highrating)
plt.title('High rating rate of each genres by time')
plt.xlabel('year')
plt.ylabel('Percentage (%)')
```



Question 2: How is the change of trending concept happend in case of horror genres?

Problem:

My favourite genre is horror. It always brings to me the interesting things. So I want to find out about the trending concept by time through the keywords which is the main contents of a film.

Solve :

Step 1: Find out top 5 most popular keywords of each period and top 5 keywords of high rating film of each time

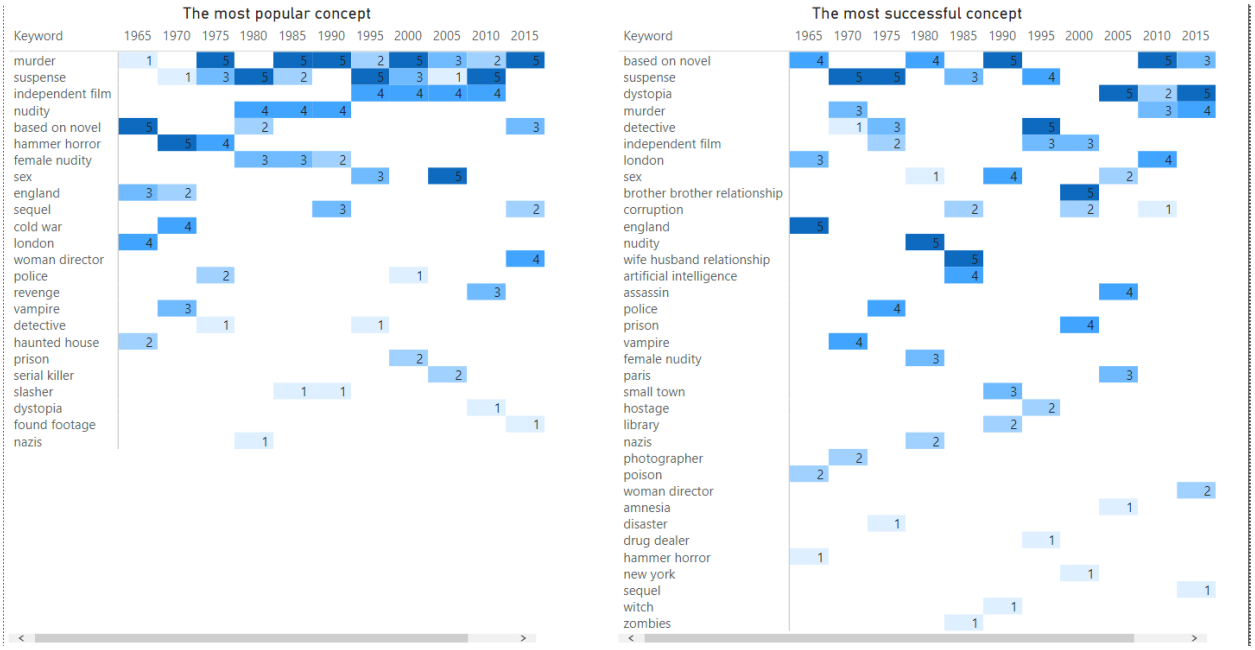
```
def horror_yn(genres):
    yn = 0
    for ele in types['Horror']:
        if ele in genres:
            yn = 1
    return yn

def Sort(sub_li):
    sub_li.sort(key = lambda x: x[1])
    return sub_li

def year_group(year):
    if year == 1960:
        return 1965
    else:
```

```
        distance = year - 1960
        period = int(distance/5)
        period = period + 1 if distance % 5 > 0 else period
        return 1960 + period*5

def add_keyword(result, group, keywords):
    keywords = keywords.split('|')
    for kw in keywords:
        if kw == 'nan':
            continue
        if group not in result.keys():
           result[group] = {kw: 1}
        else:
            if kw in result[group].keys():
                result[group][kw] += 1
            else:
                result[group][kw] = 1
    return result

def generate_top5(keywords_by_group):
    top5 = {}
    for group in keywords_by_group.keys():
        top5[group] = []
        for kw, value in keywords_by_group[group].items():
            ele = [kw, value]
            top5[group].append(ele)
        top5[group] = Sort(top5[group])
        top5[group] = top5[group][-5:]
    return top5

mostPopular_keywords_by_group = {}
highRating_keywords_by_group = {}

for index, record in data.iterrows():
    year = record['release_year']
    genres = record['genres']
    rating = record['vote_average']
    if horror_yn(str(genres)) == 0:
        continue
    group = year_group(year)
    keywords = record['keywords']

    if rating >= 6.5:
        highRating_keywords_by_group = add_keyword(highRating_keywords_by_group,
group, str(keywords))
    mostPopular_keywords_by_group = add_keyword(mostPopular_keywords_by_group,
group, str(keywords))
```

```
top5_mostPopular = generate_top5(mostPopular_keywords_by_group)
top5_highRating = generate_top5(highRating_keywords_by_group)
```

Step 2: Visualize the top 5 keywords

Python and relevant libraries don't have the chart I need, so I will use Power BI to visualize



4. Conclusions:

There are 3 main genres, they are Drama, Action/Adventure and Horror

As the below chart, we can see the number of films is increasing from year to year

But, how does the audience think about them?

As the above chart, ROE of the current period is too low if we compare with ROE in 1970 – 1980 period. But 5-10% is not bad.

There aren't too much difference among these genres. However, the Horror and Action /Adventure recently seem to do better than Drama in ROE. Maybe audience are prefering a strong feeling to a comportable moment.

And how about the rating?

Too sad, the growth of number of films doesn't help quality of film better. Actually, there are too much film have the same concept or a terrible script. But as above charts, there is a improvement of quality recently. Hope we can see more amazing films.

How is the change of trending concept happend in case of horror genres?

There are some concepts always make us excited. A beginning with “based on a true story/based on a novel” and the murders such as Micheal Myer, Jason,… are legends of this genre.

We also have some new stars like dystopia and corruption. Equilibrium, Children of men, The lobster,… is the outstanding representative.