MICROSOFT MOVIES

Author: Ruth Kamau

Overview: In this repo we are looking to recommend the movie genres with the highest rating for microsoft to adapt.

PROBLEM STATEMENT: Microsoft sees all the big companies creating original video content and they want to get in on the fun. They have decided to create a new movie studio, but they don’t know anything about creating movies. You are charged with exploring what types of films are currently doing the best at the box office. You must then translate those findings into actionable insights that the head of Microsoft's new movie studio can use to help decide what type of films to create.

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

```
# data frame
df=pd.read_csv('bom.movie_gross.csv')
df
```

1 to 25 of 3387 entries Filter

| index | title | studio | domestic_gross | foreign_gross | year |
|---|---|---|---|---|---|
| 0 | Toy Story 3 | BV | 415000000.0 | 652000000 | 2010 |
| 1 | Alice in Wonderland (2010) | BV | 334200000.0 | 691300000 | 2010 |
| 2 | Harry Potter and the Deathly Hallows Part 1 | WB | 296000000.0 | 664300000 | 2010 |
| 3 | Inception | WB | 292600000.0 | 535700000 | 2010 |
| 4 | Shrek Forever After | P/DW | 238700000.0 | 513900000 | 2010 |
| 5 | The Twilight Saga: Eclipse | Sum. | 300500000.0 | 398000000 | 2010 |
| 6 | Iron Man 2 | Par. | 312400000.0 | 311500000 | 2010 |
| 7 | Tangled | BV | 200800000.0 | 391000000 | 2010 |
| 8 | Despicable Me | Uni. | 251500000.0 | 291600000 | 2010 |
| 9 | How to Train Your Dragon | P/DW | 217600000.0 | 277300000 | 2010 |
| 10 | Clash of the Titans (2010) | WB | 163200000.0 | 330000000 | 2010 |
| 11 | The Chronicles of Narnia: The Voyage of the Dawn Treader | Fox | 104400000.0 | 311300000 | 2010 |
| 12 | The King's Speech | Wein. | 135500000.0 | 275400000 | 2010 |
| 13 | Tron Legacy | BV | 172100000.0 | 228000000 | 2010 |
| 14 | The Karate Kid | Sony | 176600000.0 | 182500000 | 2010 |
| 15 | Prince of Persia: The Sands of Time | BV | 90800000.0 | 245600000 | 2010 |
| 16 | Black Swan | FoxS | 107000000.0 | 222400000 | 2010 |
| 17 | Megamind | P/DW | 148400000.0 | 173500000 | 2010 |
| 18 | Robin Hood | Uni. | 105300000.0 | 216400000 | 2010 |
| 19 | The Last Airbender | Par. | 131800000.0 | 187900000 | 2010 |
| 20 | Little Fockers | Uni. | 148400000.0 | 162200000 | 2010 |
| 21 | Resident Evil: Afterlife | SGem | 60100000.0 | 240100000 | 2010 |
| 22 | Shutter Island | Par. | 128000000.0 | 166800000 | 2010 |
| 23 | Salt | Sony | 118300000.0 | 175200000 | 2010 |
| 24 | Sex and the City 2 | WB (NL) | 95300000.0 | 193000000 | 2010 |

Show 25 per page 1 2 10 100 130 136

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73856 entries, 0 to 73855
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   tconst         73856 non-null  object 
 1   averagerating  73856 non-null  float64
 2   numvotes       73856 non-null  int64  
dtypes: float64(1), int64(1), object(1)
memory usage: 1.7+ MB
```

```
# describe
df.describe()
```

| | averagerating | numvotes |
|---|---|---|
| **count** | 73856.000000 | 7.385600e+04 |
| **mean** | 6.332729 | 3.523662e+03 |
| **std** | 1.474978 | 3.029402e+04 |
| **min** | 1.000000 | 5.000000e+00 |
| **25%** | 5.500000 | 1.400000e+01 |
| **50%** | 6.500000 | 4.900000e+01 |
| **75%** | 7.400000 | 2.820000e+02 |
| **max** | 10.000000 | 1.841066e+06 |

```
# data frame
df=pd.read_csv('title.basics.csv')
df
```

1 to 25 of 20000 entries Filter

| index | tconst | primary_title | original_title | start_year | runtime_minutes | genres |
|---|---|---|---|---|---|---|
| 0 | tt0063540 | Sunghursh | Sunghursh | 2013 | 175.0 | Action,Crime,Drama |
| 1 | tt0066787 | One Day Before the Rainy Season | Ashad Ka Ek Din | 2019 | 114.0 | Biography,Drama |
| 2 | tt0069049 | The Other Side of the Wind | The Other Side of the Wind | 2018 | 122.0 | Drama |
| 3 | tt0069204 | Sabse Bada Sukh | Sabse Bada Sukh | 2018 | NaN | Comedy,Drama |
| 4 | tt0100275 | The Wandering Soap Opera | La Telenovela Errante | 2017 | 80.0 | Comedy,Drama,Fantasy |
| 5 | tt0111414 | A Thin Life | A Thin Life | 2018 | 75.0 | Comedy |
| 6 | tt0112502 | Bigfoot | Bigfoot | 2017 | NaN | Horror,Thriller |
| 7 | tt0137204 | Joe Finds Grace | Joe Finds Grace | 2017 | 83.0 | Adventure,Animation,Comedy |
| 8 | tt0139613 | O Silêncio | O Silêncio | 2012 | NaN | Documentary,History |
| 9 | tt0144449 | Nema aviona za Zagreb | Nema aviona za Zagreb | 2012 | 82.0 | Biography |
| 10 | tt0146592 | Pál Adrienn | Pál Adrienn | 2010 | 136.0 | Drama |
| 11 | tt0154039 | So Much for Justice! | Oda az igazság | 2010 | 100.0 | History |
| 12 | tt0159369 | Cooper and Hemingway: The True Gen | Cooper and Hemingway: The True Gen | 2013 | 180.0 | Documentary |
| 13 | tt0162942 | Children of the Green Dragon | A zöld sárkány gyermekei | 2010 | 89.0 | Drama |
| 14 | tt0170651 | T.G.M. - osvoboditel | T.G.M. - osvoboditel | 2018 | 60.0 | Documentary |
| 15 | tt0176694 | The Tragedy of Man | Az ember tragédiája | 2011 | 160.0 | Animation,Drama,History |
| 16 | tt0187902 | How Huang Fei-hong Rescued the Orphan from the Tiger's Den | How Huang Fei-hong Rescued the Orphan from the Tiger's Den | 2011 | NaN | NaN |
| 17 | tt0192528 | Heaven & Hell | Reverse Heaven | 2018 | 104.0 | Drama |
| 18 | tt0230212 | The Final Journey | The Final Journey | 2010 | 120.0 | Drama |
| 19 | tt0247643 | Los pájaros se van con la muerte | Los pájaros se van con la muerte | 2011 | 110.0 | Drama,Mystery |
| 20 | tt0249516 | Foodfight! | Foodfight! | 2012 | 91.0 | Action,Animation,Comedy |
| 21 | tt0250404 | Godfather | Godfather | 2012 | NaN | Crime,Drama |
| 22 | tt0253093 | Gangavataran | Gangavataran | 2018 | 134.0 | NaN |
| 23 | tt0255820 | Return to Babylon | Return to Babylon | 2013 | 75.0 | Biography,Comedy,Drama |
| 24 | tt0262218 | Akakis mogzauroba | Akakis mogzauroba | 2012 | 44.0 | Documentary |

Show 25 per page 1 2 10 100 700 790 800

Like what you see? Visit the data table notebook to learn more about interactive tables.

```
Warning: total number of rows (146144) exceeds max_rows (20000). Limiting to first (20000) rows.
```

```
#data frame
df=pd.read_csv('title.ratings.csv')
df
```

1 to 25 of 20000 entries Filter

| index | tconst | averagerating | numvotes |
|---|---|---|---|
| 0 | tt10356526 | 8.3 | 31 |
| 1 | tt10384606 | 8.9 | 559 |
| 2 | tt1042974 | 6.4 | 20 |
| 3 | tt1043726 | 4.2 | 50352 |
| 4 | tt1060240 | 6.5 | 21 |
| 5 | tt1069246 | 6.2 | 326 |
| 6 | tt1094666 | 7.0 | 1613 |
| 7 | tt1130982 | 6.4 | 571 |
| 8 | tt1156528 | 7.2 | 265 |
| 9 | tt1161457 | 4.2 | 148 |
| 10 | tt1171222 | 5.1 | 8296 |
| 11 | tt1174693 | 5.8 | 2381 |
| 12 | tt1181840 | 7.0 | 5494 |
| 13 | tt1193623 | 8.0 | 5 |
| 14 | tt1199588 | 5.5 | 74 |
| 15 | tt1204784 | 5.8 | 6 |
| 16 | tt1210166 | 7.6 | 326657 |
| 17 | tt1212419 | 6.5 | 87288 |
| 18 | tt1220911 | 5.0 | 941 |
| 19 | tt1229238 | 7.4 | 428142 |
| 20 | tt1232829 | 7.2 | 477771 |
| 21 | tt1235548 | 6.6 | 2725 |

```
# Merge dataframes
```

Show 25 per page

```
#merge
df1= pd.read_csv('title.basics.csv')
df2 = pd.read_csv('title.ratings.csv')
df3=pd.merge(df1,df2)
print(df3)
```

```
          tconst                    primary_title              original_title  \
0      tt0063540                        Sunghursh                   Sunghursh
1      tt0066787  One Day Before the Rainy Season             Ashad Ka Ek Din
2      tt0069049       The Other Side of the Wind  The Other Side of the Wind
3      tt0069204                  Sabse Bada Sukh             Sabse Bada Sukh
4      tt0100275         The Wandering Soap Opera       La Telenovela Errante
...          ...                              ...                         ...
73851  tt9913084                 Diabolik sono io            Diabolik sono io
73852  tt9914286                Sokagin Çocuklari           Sokagin Çocuklari
73853  tt9914642                        Albatross                   Albatross
73854  tt9914942       La vida sense la Sara Amat  La vida sense la Sara Amat
73855  tt9916160                       Drømmeland                  Drømmeland

       start_year  runtime_minutes                genres  averagerating  \
0            2013            175.0    Action,Crime,Drama            7.0
1            2019            114.0       Biography,Drama            7.2
2            2018            122.0                 Drama            6.9
3            2018              NaN          Comedy,Drama            6.1
4            2017             80.0  Comedy,Drama,Fantasy            6.5
...           ...              ...                   ...            ...
73851        2019             75.0           Documentary            6.2
73852        2019             98.0          Drama,Family            8.7
73853        2017              NaN           Documentary            8.5
73854        2019              NaN                   NaN            6.6
73855        2019             72.0           Documentary            6.5

       numvotes
0            77
1            43
2          4517
3            13
4           119
...         ...
73851         6
73852       136
73853         8
73854         5
73855        11

[73856 rows x 8 columns]
```

```
#genre with highest rating
df=df3
df.head()
```

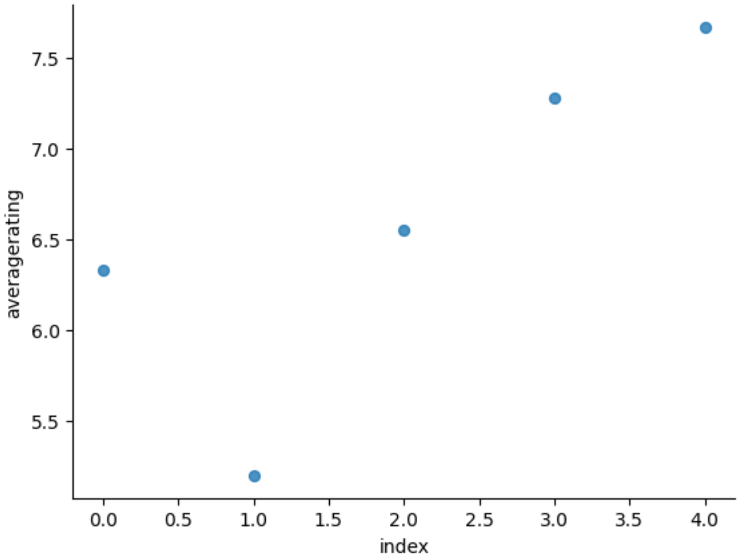
| | tconst | primary_title | original_title | start_year | runtime_minutes | genres | averagerating | numvotes |
|---|---|---|---|---|---|---|---|---|
| **0** | tt0063540 | Sunghursh | Sunghursh | 2013 | 175.0 | Action,Crime,Drama | 7.0 | 77 |
| **1** | tt0066787 | One Day Before the Rainy Season | Ashad Ka Ek Din | 2019 | 114.0 | Biography,Drama | 7.2 | 43 |
| **2** | tt0069049 | The Other Side of the Wind | The Other Side of the Wind | 2018 | 122.0 | Drama | 6.9 | 4517 |
| **3** | tt0069204 | Sabse Bada Sukh | Sabse Bada Sukh | 2018 | NaN | Comedy,Drama | 6.1 | 13 |
| **4** | tt0100275 | The Wandering Soap Opera | La Telenovela Errante | 2017 | 80.0 | Comedy,Drama,Fantasy | 6.5 | 119 |

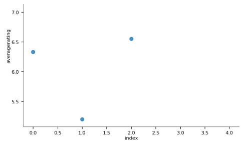
```
grouped = df.groupby(['numvotes','genres'])['averagerating'].mean().reset_index()
grouped.head()
```

1 to 5 of 5 entries Filter

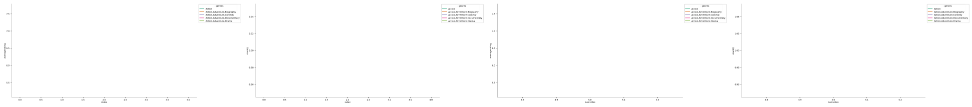
| index | numvotes | genres | averagerating |
|---|---|---|---|

```
#genre vs average rating
from matplotlib import pyplot as plt
_df_10.plot(kind='scatter', x='index', y='averagerating', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

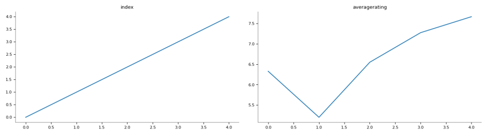




### Time series



### Values



### Faceted distributions

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.