Movie Review Discrepancy Analysis

DA514 Project

KETUPATI SWARGIARY 244161015

PROJECT OVERVIEW

- Is there a conflict of interest for a website that both sells movie tickets and displays review ratings?
- Do they have a bias towards rating movies higher than they should be rated?
- Can we answer this with data analysis?

- FiveThirtyEight 2015 article ?

FiveThirtyEight

Politics Sports Science Podcasts Video Interactives

This is an archived site and is no longer being updated. New 538 articles can be foun

OCT. 15, 2015, AT 9:52 AM

Be Suspicious Of Online Movie Ratings, Especially Fandango's

By Walt Hickey

Filed under Movies

Get the data on GitHub

"Ted 2," "Avengers: Age of Ultron," and "Fantastic Four"

You were excited for the date: dinner and a movie. Your date picked a restaurant — "It got five stars on Yelp!" — but the movie was up to you. So you checked out what was playing and bought the tickets on Fandango's website. You decided to check out "Fantastic Four," and even though you

Wikipedia The Free Encyclopedia

Donate Create account Log in

Fandango Media

15 languag

Read Edit View history To

Contents

(Top)
History
Services
Competition
Controversies
Video on demand
See also
References
External links

dia

merican ticketing company that
site and their mobile app. It also
erly known as Vudu), a streaming
g service, as well as Rotten
vision and streaming media

ounded Fandango.[6]

In 2003, Fandango secured $15 million in funding from venture capitalists Technology Crossover Ventures.[7] Fandango was privately held.[8] Then-owners included exhibition chains (Loews Cineplex Entertainment, Regal Cinemas, Carmike Cinemas, Cinemark Theatres, General Cinema Theatres, Edwards Theatres and Century Theatres[8]) and venture capital firms (*Accretive Technology Partners* and *General Atlantic Partners*).

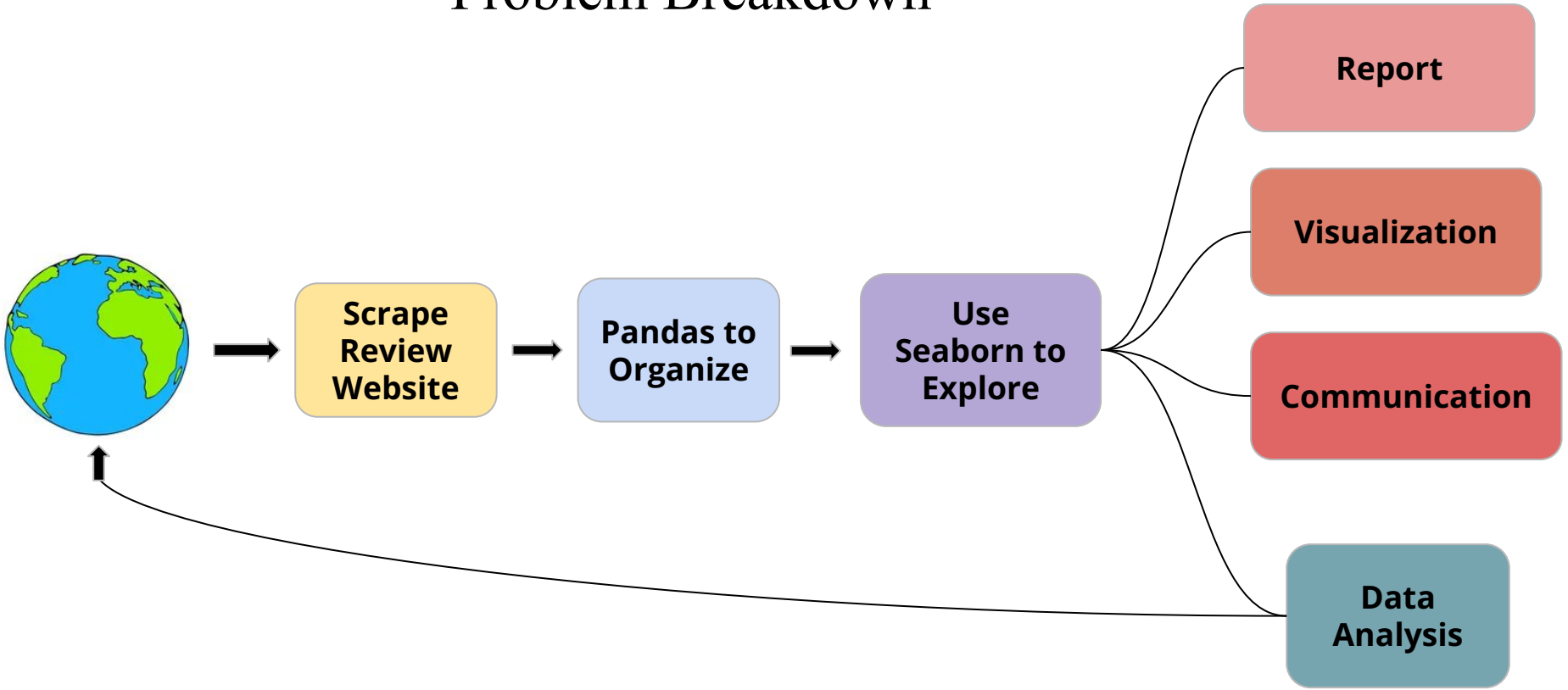
On April 11, 2007, Comcast acquired Fandango, with plans to

Fandango Media, LLC

Fandango headquarters in Beverly Hill

Formerly	ticketmakers.com (2000–2
Company type	Joint venture
Founded	April 27, 2000; 24 years a
Founder	James Michael Cline

Problem Breakdown



Overview of Dataset

- Two data sets
  - fandango_scrape.csv
  - fandango_score_comparison.csv
- Fandango has two ratings:
  - STARS (displayed on website)
  - RATING (numerical)
-

```
[113]: import pandas as pd
       import matplotlib.pyplot as plt
       import seaborn as sns

[114]: fandango = pd.read_csv("fandango_scrape.csv")
       fandango.head()
```

[114]:

	FILM	STARS	RATING	VOTES
0	Fifty Shades of Grey (2015)	4.0	3.9	34846
1	Jurassic World (2015)	4.5	4.5	34390
2	American Sniper (2015)	5.0	4.8	34085
3	Furious 7 (2015)	5.0	4.8	33538
4	Inside Out (2015)	4.5	4.5	15749

```
[116]: fandango.describe() #to see any statistical p
```

[116]:

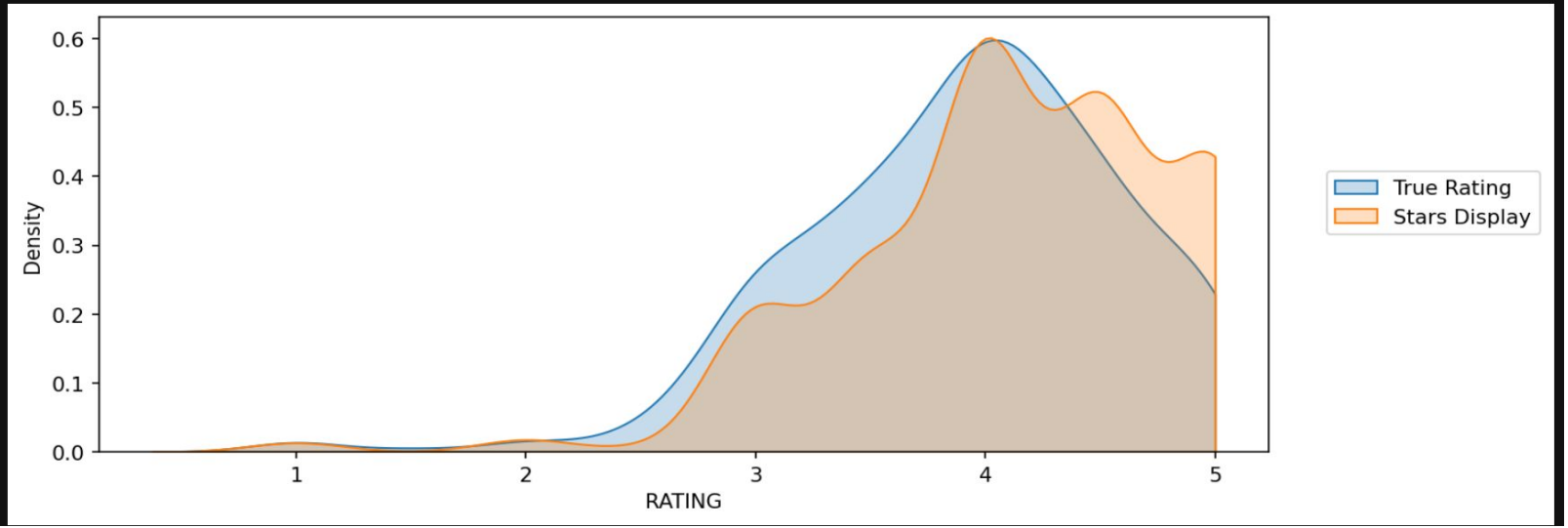
	STARS	RATING	VOTES
count	510.000000	510.000000	510.000000
mean	3.532353	3.351765	1134.364706
std	1.585616	1.512628	3809.952176
min	0.000000	0.000000	0.000000
25%	3.500000	3.025000	2.000000
50%	4.000000	3.800000	17.500000
75%	4.500000	4.300000	183.500000
max	5.000000	5.000000	34846.000000

Initial Statistical Property

Visualization of True Rating and Stars displayed

```
[124]: #plot b/w True rating and Stars Displayed
       plt.figure(figsize=(10,4),dpi=150)
       sns.kdeplot(data=fan_reviewed,x='RATING',clip=[0,5],fill=True,label='True Rating')
       sns.kdeplot(data=fan_reviewed,x='STARS',clip=[0,5],fill=True,label='Stars Display')
       plt.legend(loc=(1.05,0.5))
```

```
[124]: <matplotlib.legend.Legend at 0x15455a04750>
```



Data set and Statistical Property of other websites

[156]:
```
#COMPARISON OF FANDANGO RATINGS TO OTHER SITES (Rotten Tomatoes,Metacritic,IMDB)
all_sites = pd.read_csv("fandango_score_comparison.csv")
all_sites.head()
```

[156]:

	FILM	RottenTomatoes	RottenTomatoes_User	Metacritic	Metacritic_User	IMDB	Fandango_Stars	Fandango_Ratingvalue	RT_norm
0	Avengers: Age of Ultron (2015)	74	86	66	7.1	7.8	5.0	4.5	3.70
1	Cinderella (2015)	85	80	67	7.5	7.1	5.0	4.5	4.25
2	Ant-Man (2015)	80	90	64	8.1	7.8	5.0	4.5	4.00
3	Do You Believe? (2015)	18	84	22	4.7	5.4	5.0	4.5	0.90

Data set and Statistical Property of other websites

```
[129]: all_sites.describe()
```

[129]:

	RottenTomatoes	RottenTomatoes_User	Metacritic	Metacritic_User	IMDB	Fandango_Stars	Fandango_Ratingvalue	RT_norm
count	146.000000	146.000000	146.000000	146.000000	146.000000	146.000000	146.000000	146.000000
mean	60.849315	63.876712	58.808219	6.519178	6.736986	4.089041	3.845205	3.042466
std	30.168799	20.024430	19.517389	1.510712	0.958736	0.540386	0.502831	1.508440
min	5.000000	20.000000	13.000000	2.400000	4.000000	3.000000	2.700000	0.250000
25%	31.250000	50.000000	43.500000	5.700000	6.300000	3.500000	3.500000	1.562500
50%	63.500000	66.500000	59.000000	6.850000	6.900000	4.000000	3.900000	3.175000
75%	89.000000	81.000000	75.000000	7.500000	7.400000	4.500000	4.200000	4.450000
max	100.000000	94.000000	94.000000	9.600000	8.600000	5.000000	4.800000	5.000000

8 rows × 21 columns

Normalizing and Filtering the Data Set

[141]:
```
#new normalized columns for all ratings so they match up with the 0-5 star range shown on Fandango.
import numpy as np
df['RT_Norm']=np.round(df['RottenTomatoes']/20,1)
df['RTU_Norm']=np.round(df['RottenTomatoes_User']/20,1)
df['Meta_Norm']=np.round(df['Metacritic']/20,1)
df['Meta_U_Norm']=np.round(df['Metacritic_User']/2,1)
df['IMDB_Norm']=np.round(df['IMDB']/2,1)
df.head()
```

[141]:

	FILM	STARS	RATING	VOTES	YEAR	RottenTomatoes	RottenTomatoes_User	Metacritic	Metacritic_User	IMDB	...	Metacritic_user_vote_count	IMD
0	Fifty Shades of Grey (2015)	4.0	3.9	34846	2015	25	42	46	3.2	4.2	...	778	
1	Jurassic World (2015)	4.5	4.5	34390	2015	71	81	59	7.0	7.3	...	1281	

Normalized Data

```
[143]: norm_scores = df[['STARS', 'RATING','RT_Norm','RTU_Norm','Meta_Norm','Meta_U_Norm','IMDB_Norm']]
       norm_scores.head()
```

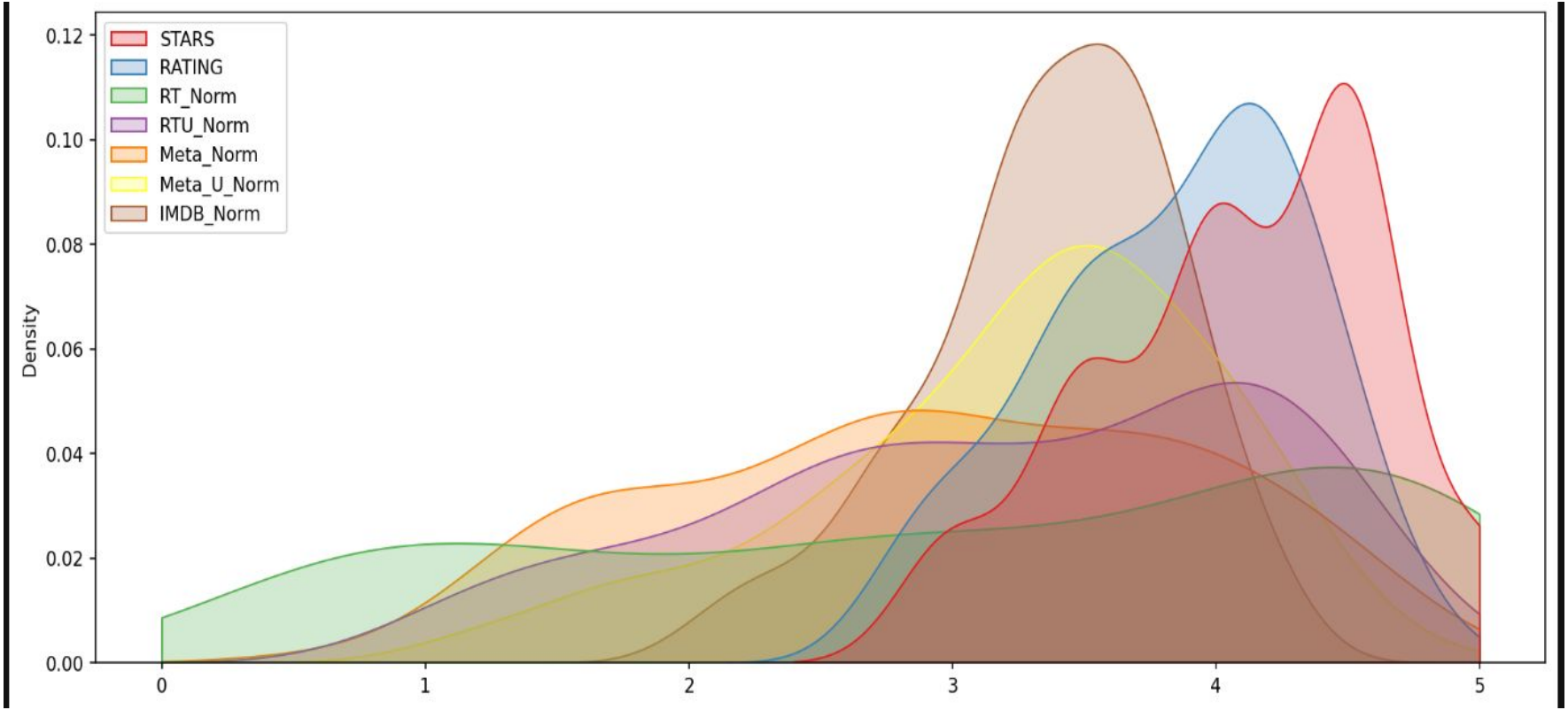
[143]:

	STARS	RATING	RT_Norm	RTU_Norm	Meta_Norm	Meta_U_Norm	IMDB_Norm
0	4.0	3.9	1.2	2.1	2.3	1.6	2.1
1	4.5	4.5	3.6	4.0	3.0	3.5	3.6
2	5.0	4.8	3.6	4.2	3.6	3.3	3.7
3	5.0	4.8	4.0	4.2	3.4	3.4	3.7
4	4.5	4.5	4.9	4.5	4.7	4.4	4.3

Comparison of Distributions Across all sites

```
•[148…  #Comparing Distribution of Scores Across Sites
        def move_legend(ax, new_loc, **kws):
            old_legend = ax.legend()
            handles = old_legend.legend_handles
            labels = [t.get_text() for t in old_legend.get_texts()]
            title = old_legend.get_title().get_text()
            ax.legend(handles, labels, loc=new_loc, title=title, **kws)
        fig, ax = plt.subplots(figsize=(15,6),dpi=150)
        sns.kdeplot(data=norm_scores,clip=[0,5],fill=True,palette='Set1',ax=ax)
        move_legend(ax, "upper left")
```

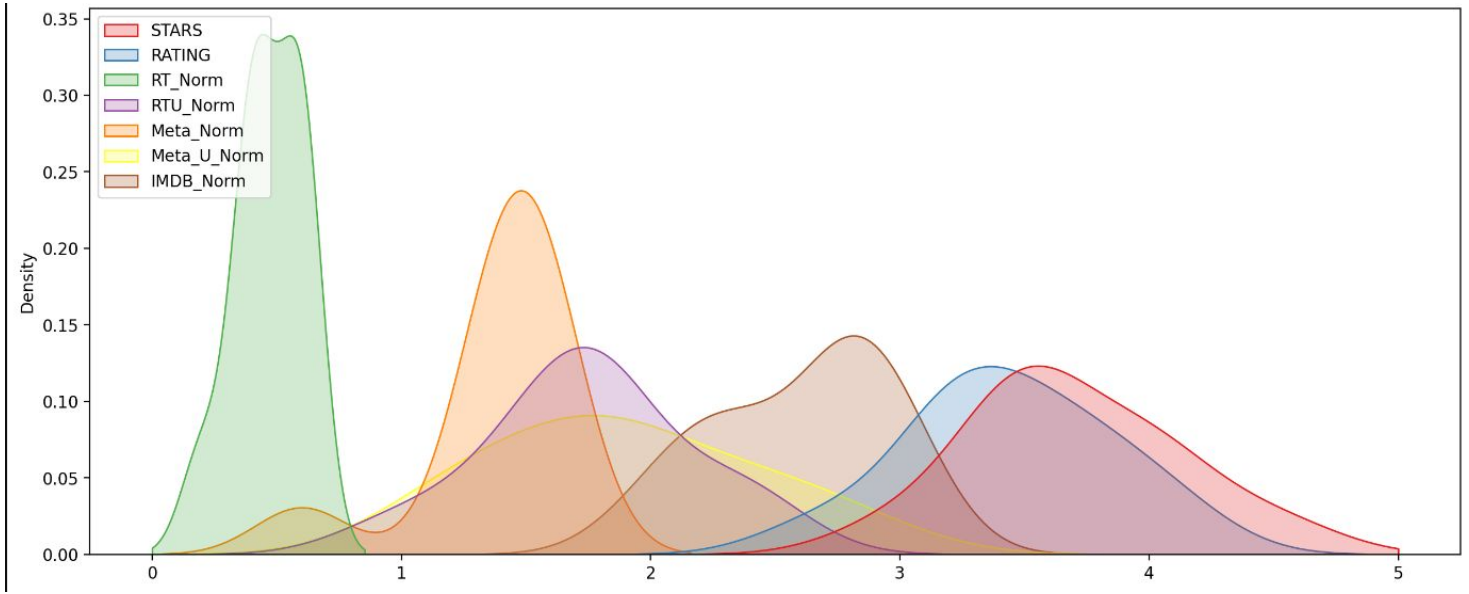
Comparison of Distributions Across all sites



Distribution of top 10 Worst Movies

```
[154]: #Top 10 worst films in the list.
       norm_films = df[['FILM','STARS', 'RATING', 'RT_Norm', 'RTU_Norm', 'Meta_Norm', 'Meta_U_Norm',
              'IMDB_Norm']]
       worst_films = norm_films.nsmallest(10,'RT_Norm')

[155]: #Distribution of ratings across all sites for the top 10 worst movies.
       fig, ax = plt.subplots(figsize=(15,6),dpi=200)
       sns.kdeplot(data=worst_films,clip=[0,5],fill=True,palette='Set1',ax=ax)
       move_legend(ax, "upper left")
```



THANK YOU