Movie Recommendation Application

Ryan S. Shaw

Northwest Missouri State University, Maryville MO 64468, USA
S546850@nwmissouri.edu

Abstract. Keywords: data analytics · movies · Python · web scraping

1 Introduction

I plan to work in the domain of web scraping. Primarily because it is the most interesting to me but also because I feel I did not get to fully explore web scraping in the web scraping course due to time constraints.

I would like to scrape my data from the online website IMDB primarily. I may also source data from Rotten Tomatoes.

I intend to solve the problem of trying to decide which movies to watch. I feel like this is something everyone struggles with at some point. Scrolling through endless movies on Netflix or some other streaming service and not knowing what to choose.

Steps taken would be as follows:

1. Scrape the data from the previously mentioned websites.
2. Clean/organize the data into a usable format.
3. Create a Python script that requests user information (name, birthdate, gender, favorite genre).
4. Create a model that can recommend some movies that the user may like based on their information and the movie rating.

Python will be a huge component for me in this project. I will need it to for just about every aspect of this project.

2 Data Sourcing

I used Python and web scraping (specifically the Beautiful Soup Module) to gather my data from the IMDB website. The data is in HTML format being written intitially for a webpage and collected using Python. The attributes that will be gathered are: Title, Release Year, Rating, Runtime, Genre, Metascore, Movie Description and Votes. IMDB organizes movies by genre, so moving forward I would like to implement a method for users to search movie recommendations by genre.

3 Data Cleaning

First, I wanted to take a look at the output data to get an idea of how my data needed cleaned:

title	1.
year	-2023
certificate	PG-13
time	154 min
genre	Action, Adventure
rating	6.9
metascore	58
simple_de	Archaeologist Indiana Jones races against time to retrieve a legendary artifact that can change the course of history.
votes	Votes:
title	2.
year	-2023
certificate	PG-13
time	131 min
genre	Action, Biography, Drama
rating	8.3
metascore	43
simple_de	The incredible true story of a former government agent turned vigilante who embarks on a dangerous mission to rescue hundreds of children from sex traffickers.

Fig. 1. Output CSV file before data cleaning.

As can be seen in Fig. 1, the output CSV file generated by my code is not formatted very well and difficult to read. The original code I used to output my information to a CSV file can be seen in Fig. 2 and the updated code in Fig. 3.

```
49
50          movie_data.append(data)
51
52          with open(r'c:\Users\User\Desktop\Capstone\Module 2\module2.csv', 'w') as csv_file:
53              writer = csv.writer(csv_file)
54              for y in movie_data:
55                  for key, value in y.items():
56                      writer.writerow([key, value])
57
58
59
60
```

Fig. 2. Previous code used to write the CSV file.

```
movie_data.append(data)

with open(r'C:\Users\User\Desktop\Capstone\Module 3\ module3.csv', 'w', newline='') as csv_file:

    for y in movie_data:


        writer = csv.DictWriter(csv_file, fieldnames=y.keys())

        writer.writerow(y)
```

Fig. 3. New code to write to CSV file.

I changed the Writer method to DictWriter since it is specifically made to handle writing dictionary objects to CSV which is what I am doing here. This keeps each movie record on a line instead of printing each attribute of every movie on a new line.

I also added the newline argument. This tells Python exactly how I want it to handle new lines. As you can see in Fig. 1, it was skipping a line after each record. It no longer does this thanks to the newline argument being specified.

These were the biggest changes made using Python. The rest I was able to easily accomplish using Microsoft Excel as you can see in Fig. 4. I added a title row which shows the names of each attribute, bolded. I also centered all the data and expanded the cells to see the data better. I also deleted the Year column completely, as it is not important for my project and IMDB includes this information in the movie titles, making this redundant. This left me with 8 attributes and had no missing values that needed cleaning.

Movie Title	Rating	Runtime	Genres	IMDB User Rating	Metascore Rating	Description	Number of Votes
1. Indiana Jones and the Dial of Destiny (2023)	PG-13	154 min	Action, Adventure	6.9	58 Metascore	Archaeologist Indiana Jones races against time	Votes: 71,063
2. Sound of Freedom (2023)	PG-13	131 min	Action, Biography, Drama	8.2	43 Metascore	The incredible true story of a former governme	Votes: 22,491
3. Spider-Man: Across the Spider-Verse (2023)	PG	140 min	Animation, Action, Adventure	8.9	86 Metascore	Miles Morales catapults across the Multiverse,	Votes: 182,518
4. Mission: Impossible - Dead Reckoning Part One (2023)	PG-13	163 min	Action, Adventure, Thriller	8.1	81 Metascore	Ethan Hunt and his IMF team must track down	Votes: 45,855
5. The Flash (2023)	PG-13	144 min	Action, Adventure, Fantasy	7.1	56 Metascore	Barry Allen uses his super speed to change the	Votes: 88,374
6. Nimona (2023)	PG	101 min	Animation, Action, Adventure	7.7	75 Metascore	When a knight in a futuristic medieval world is	Votes: 15,615
7. Extraction II (2023)	R	122 min	Action, Thriller	7.1	57 Metascore	After barely surviving his grievous wounds fro	Votes: 96,544
8. Guardians of the Galaxy Vol. 3	PG-13	150 min	Action, Adventure, Comedy	8.1	64 Metascore	Still reeling from the loss of Gamora, Peter Qu	Votes: 213,698

Fig. 4. New output CSV file after data cleaning.

This leaves me with the following attributes:

1. Movie Title (Dependent Variable): The title of the movies.
2. Rating: (Independent Variable): Movie rating (E.g. G, PG - 13, R, etc.).
3. Runtime: Length of movie.
4. Genres (Independent Variable): Movie genre according to IMDB.
5. IMDB User Rating (Independent Variable): Average rating (out of 10 possible) given by IMDB users.
6. Metascore Rating (Independent Variable): Critic rating (out of 100 possible) assigned by movie critics.
7. Description: Movie description.
8. Number of Votes: Number of votes by IMDB users for that movie.

4 Analysis

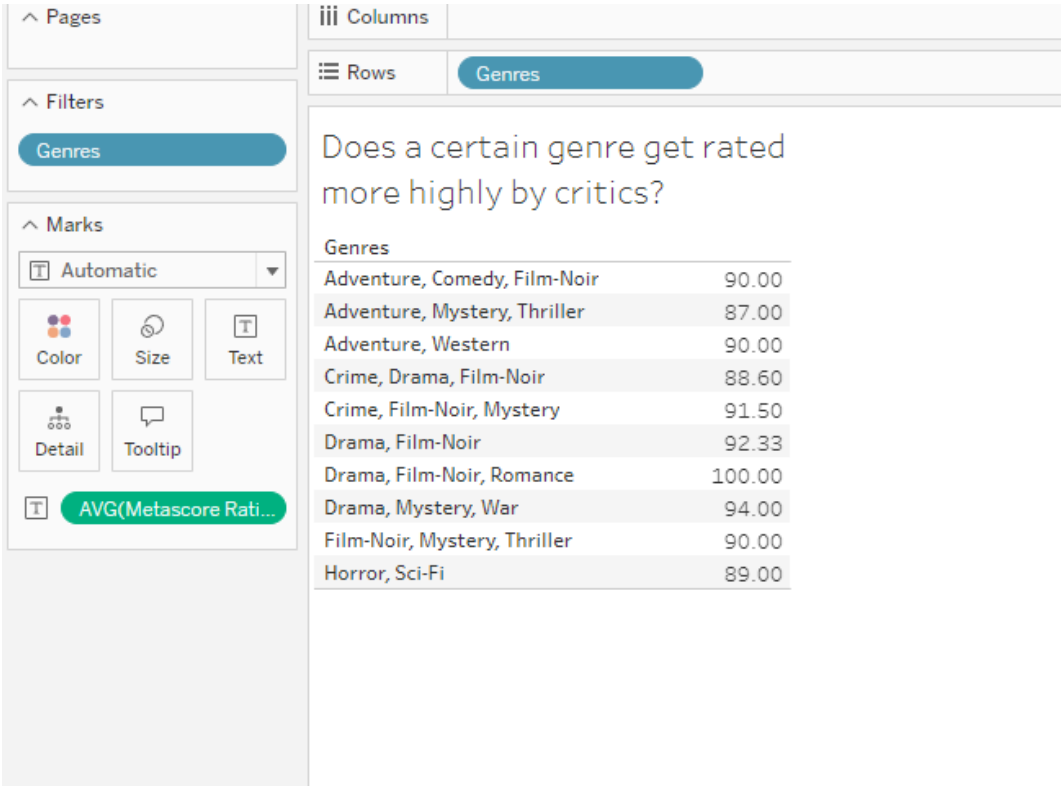
Exploratory data analysis (EDA), refers to the act of investigating your dataset in order to uncoer patterns, test hypotheses or assumptions, and discover anomalies. This step is essential for any data analytics project as helps spot missing or incorrect data, determine the most important values, find patterns or anomalies, and prove/disprove hypotheses.

There are four primary data analysis techniques:

1. Univariate Non-Graphical: This technique only involves 1 variable and does not deal with relationships.
2. Univariate Graphical: Also deals with only 1 variable and utilizes graphical methods to provide a fuller picture of the data.

3. Multivariate Non-Graphical: Data has multiple variables. Shows the relationships between data using statistics or cross-tabulation.
4. Multivariate Graphical: Data has multiple variables. Uses graphics to show relationships between the data. (This is the technique I used in my own EDA.)

I came up with four questions that I wanted to answer through data exploration in Tableau. The first question I wanted to know was: Is/are there movie genre(s) that get rated more highly by critics?



Genres	
Adventure, Comedy, Film-Noir	90.00
Adventure, Mystery, Thriller	87.00
Adventure, Western	90.00
Crime, Drama, Film-Noir	88.60
Crime, Film-Noir, Mystery	91.50
Drama, Film-Noir	92.33
Drama, Film-Noir, Romance	100.00
Drama, Mystery, War	94.00
Film-Noir, Mystery, Thriller	90.00
Horror, Sci-Fi	89.00

Fig. 5. Table showing the highest rated genres by critics.

As you can see in Figure 5, Drama, Adventure, Film-Noir, and Crime feature heavily in the top 10 highest rated genres by critic metascore on IMDB.

Next, is/are there movie genre(s) that get rated more highly by IMDB users?

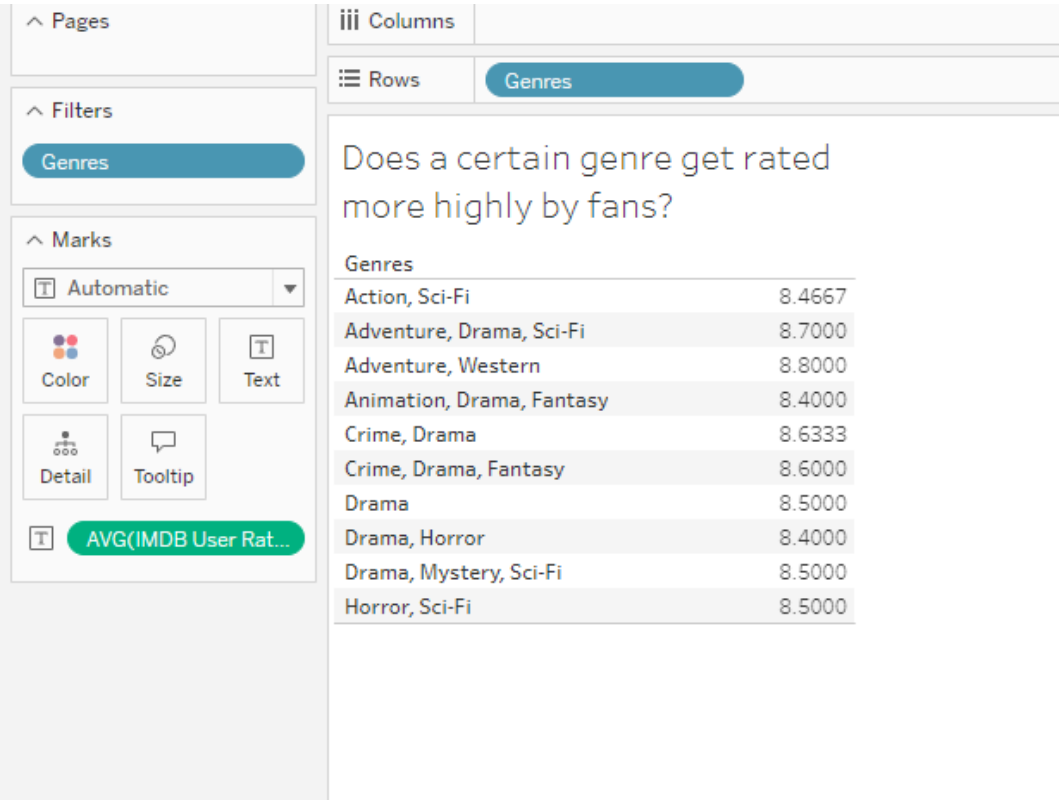


Fig. 6. Table showing the highest rated genres by IMDB users.

As shown in Figure 6, Drama features very heavily in the top 10 highest rated genres by IMDB user score, with Crime, Adventure, Fantasy, and Sci-Fi among some others, also being mentioned several times.

Are there any movies that show a large discrepancy in ratings between critics and fans?

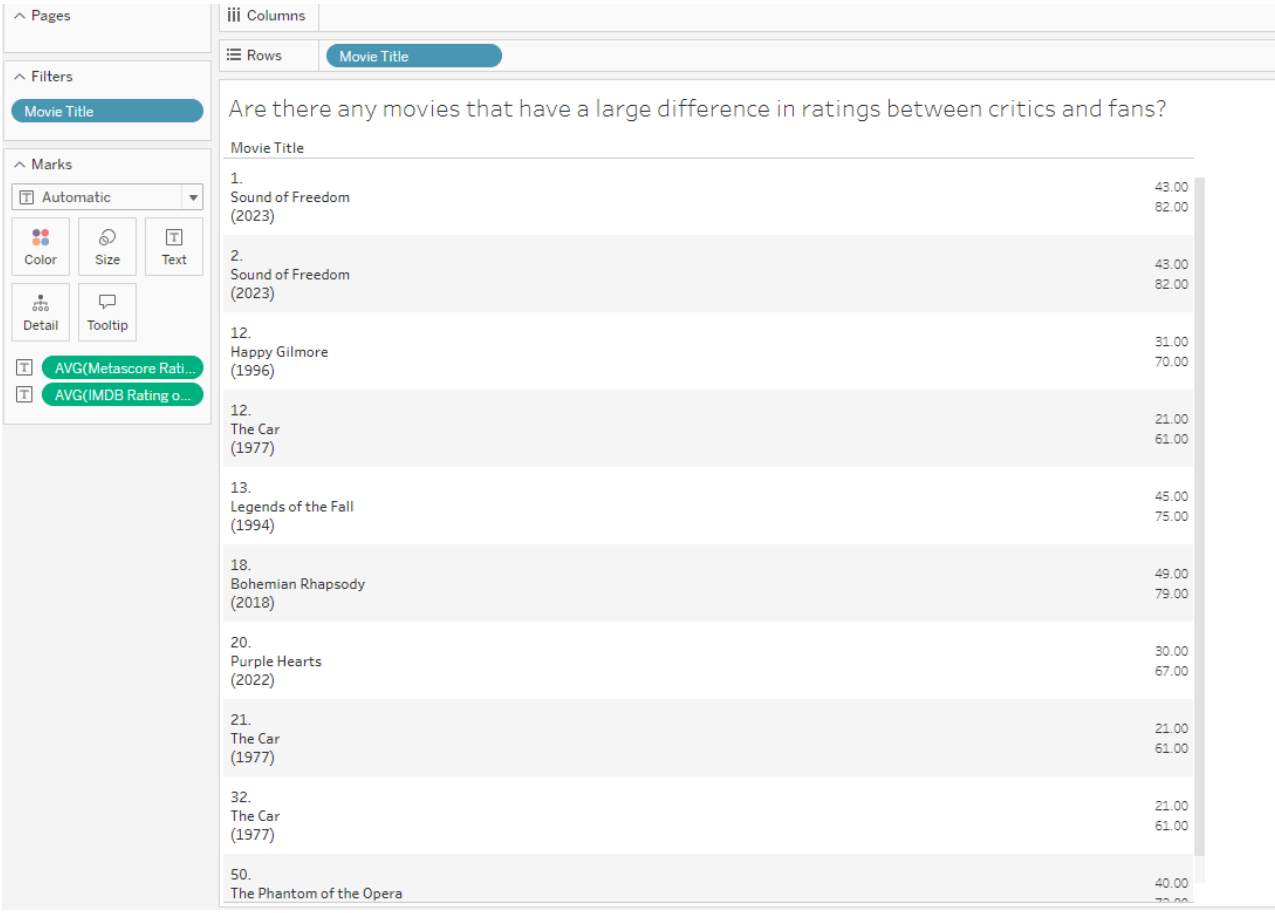


Fig. 7. Comparison of IMDB user rating vs critic rating.

Figure 7 shows the 10 movies with the largest difference in IMDB user rating and critic rating. Since IMDB user rating is only on a 10 point scale and metascore rating is on a 100 point scale, I first had to multiply all the user ratings by 10. This way I could make a more direct comparison between the two ratings. The top number on the right is the metascore rating given by critics and the bottom number is the IMDB user rating. This shows that casual movie watchers think quite highly of these films despite the poor critical perception.

Another thing to notice are the duplicate movie titles. This is because the movies are listed by genre and because a movie can belong to multiple genres, they can be listed several times. This is a good example of what can be revealed through proper EDA. In this case, I will need to further clean my data in order to remove any duplicate values.

Finally, does a certain movie rating (i.e., PG, PG - 13, R, etc.) get rated more highly than others?

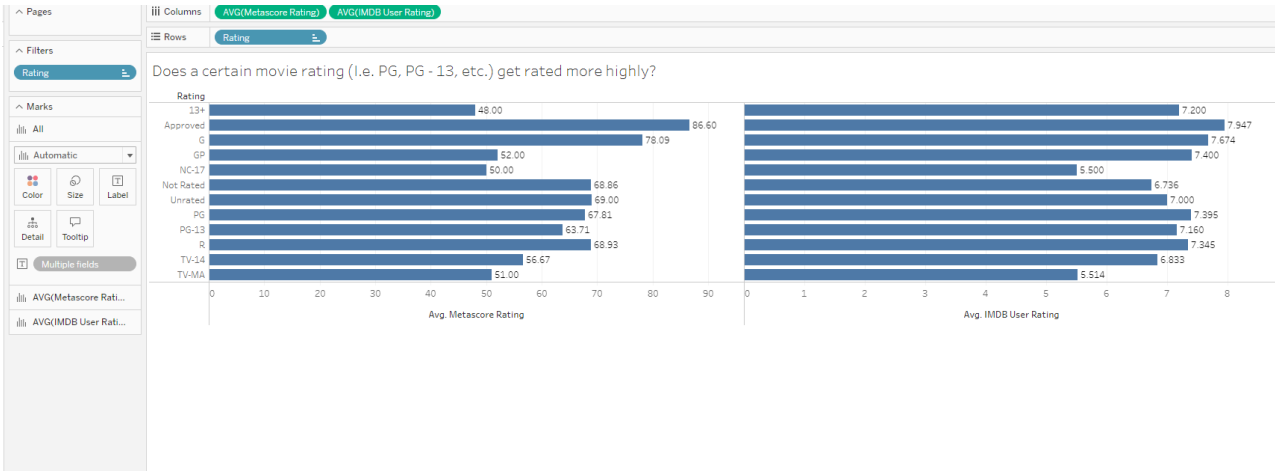


Fig. 8. Bar chart showing the average IMDB score for each movie rating.

Figure 8 shows that the highest rated film rating for both critics and IMDB users is "Approved". Approved is actually a rating given to movies prior to 1968 if they were deemed "moral". I suspect this data is an outlier as there would be much fewer movies produced before 1968 and even fewer still being watched today. Of those that are still being watched, they would have to be good in order to stand the test of time. The next highest would be G rated films.

5 Results

6 Limitations

7 Conclusion

[]

References

1. Imdb: Ratings, reviews, and where to watch the best movies and tv, https://www.imdb.com/
2. Rotten tomatoes: Movies — tv shows, https://www.rottentomatoes.com/