

DEVELOPMENT PART-2

PREDICTION IMDB SCORE

INTRODUCTION: IMDb is a platform that provides a vast collection of data related to films, [TV series](#), and other media content. It offers users the ability to explore details about movies and shows, such as cast and crew information, release dates, plot summaries, trivia, user reviews, ratings, and much more. Whether you're a casual moviegoer or a dedicated film enthusiast, IMDb offers a wealth of information to satisfy your curiosity.

MDb's Role in the Entertainment Industry

Importance and Influence

1. **Information Hub:** IMDb is a central repository of data that offers an unparalleled collection of information about movies, TV shows, actors, directors, and crew members. [Filmmakers](#), journalists, researchers, and enthusiasts rely on IMDb for accurate and comprehensive details about productions.
2. **Audience Engagement:** IMDb provides a platform for viewers to rate, review, and discuss movies and TV shows. User-generated content fosters engagement, allowing audiences to express opinions, form communities, and engage in conversations around their favorite media.
3. **Talent Discovery:** As a database accessible to industry professionals, IMDb serves as a tool for discovering new and established talent. Actors, directors, and crew members can showcase their work, facilitating networking and collaboration opportunities.

User-Generated Content on IMDb

User Reviews and Ratings

4. **Personal Reviews:** Users can write detailed reviews for movies and TV shows, sharing their thoughts, [analyses](#), and [recommendations](#) with the community.

5. **Star Ratings:** IMDb allows users to rate movies and TV shows on a scale of 1 to 10 stars, contributing to the overall rating and helping others gauge audience reception.

Contributions to Data Accuracy

6. **Edit and Correct Information:** Users have the ability to edit and correct certain information on IMDb. This collaborative effort enhances the accuracy and completeness of the database.
7. **Trivia and Quotes:** Enthusiasts can add interesting trivia, memorable quotes, and connections between movies, contributing to the platform's depth of information.

Community and Forums

8. **Message Boards (Formerly):** IMDb used to host message boards where users could engage in discussions about movies, actors, and various topics related to the entertainment industry.
9. **Community Contributions:** Users can participate in discussions, [ask questions](#), and share their expertise within the IMDb community, connecting with others who share similar interests.

IMDbPro: The Professional Version

Benefits for Industry Professionals

-
- 10. **Contact Information:** IMDbPro provides access to contact details of industry insiders, including actors, directors, producers, agents, and executives. This facilitates [direct communication](#) and collaboration.
- 11. **Casting Information:** IMDbPro offers insights into casting details for upcoming roles, enabling actors and casting directors to find suitable talent for projects.
- 12. **Box Office Data:** IMDbPro provides box office data, including opening weekend grosses and historical performance, helping professionals gauge the success of different titles.

Advanced Features and Tools

-
- 13. **Starmeter Rankings:** IMDbPro subscribers can see their "Starmeter" rankings, which reflect their popularity and visibility within the [entertainment industry](#).
- 14. **Enhanced Profiles:** IMDbPro users have the option to add additional information and media to their profiles, making them more informative and engaging for potential collaborators.
- 15. **Messaging:** The platform enables direct messaging between professionals, streamlining communication for networking and collaboration purposes.

```
import numpy as np import pandas as pd
```

In [2]:

```
data = pd.read_csv('movie_metadata.csv') data.head()
```

```
data.dropna(axis=0,how='any')
```

Out[5]:

```
data.dropna(axis=0,how='any')
```

Out[5]:

	color	director_name	num_critic_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross	genres	...	num_user_reviews	language	country	content_rating	budget	title_year	actor_2_facebook_likes	imdb_score	aspect_ratio	movie_facebook_likes
0	Color	James Cameron	7230	1780	0.850	Joel David Moore	10000	7600508470	Action Adventure Fantasy Sci-Fi	...	30540	English	USA	PG-13	237000000	2009	9360	7.9	1.78	33000	

										i-Fi											
1	Color	Greene Verbinski	302.0	169.0	563.0	100.0	Orlando Bloom	4000.0	30944152.0	Action Adventure Fantasy	...	1238.0	English	USA	PG-13	3000000.0	2007.0	5000.0	7.1	2.35	0
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	20074175.0	Action Adventure Thriller	...	994.0	English	UK	PG-13	2450000.0	2015.0	393.0	6.8	2.35	85000
3	Color	Christopher Nolan	813.0	164.0	2200.0	2300.0	Christian Bale														

```

datadf = pd.read_csv('movie_metadata.csv') #color, duration, actor_3_fb_likes datadf =
datadf.drop(datadf.columns[[0,1,3,5,6,8,10,11,14,15,16,17,19,20,21,23,26]],axis=1)

```

```
datadf.head()
```

```
datadf = datadf.replace(0,float("NaN"))
```

```
datadf.head() print len(datadf)
```

```
datadf = datadf.dropna(axis=0,how='any') print len(datadf)
```

```
1990
```

```
:
```

```
datadf.head()
```

```
datadf = datadf.drop(datadf.columns[[3]],axis=1)
```

```
datadf = datadf.round(0)
```

```
datadf = datadf.astype(int)
```

```
datadf.head()
```

```
np.array(datadf)
```

Out[29]:

```
array([[ 813, 22000, 27000, ..., 23000,      8, 164000],
       [ 462,   475,   640, ...,   632,      7,  24000],
       [ 324,    15,   799, ...,   553,      8,  29000],
       ...,
       [   13,   120,   785, ...,    98,      5,    424],
       [  143,   291,   291, ...,    45,      7,  19000],
       [   43,    16,    86, ...,    23,      7,   456]])
```

In [34]:

```
type(datadf['budget'])
```

Out[34]:

```
pandas.core.series.Series
```

In [37]:

```
type(datadf.iloc[:,6])
```

Out[37]:

```
pandas.core.series.Series
```

In [42]:

```
datadf = pd.read_csv('movie_metadata_filtered_aftercsv.csv') #separate classes and stuffs y =  
np.array(datadf['imdb_score']) X = np.array(datadf.drop(datadf.columns[[0,8]],axis=1))
```

In [43]:

```
X
```

```
array([[ 813, 22000, 27000, ..., 250000000,      8,  
        164000],  
       [ 462,   475,   640, ..., 263700000,      6,
```

```

[ 24000],
  324,    15,    799, ..., 260000000,    7,
 29000],
...,
[   13,   120,   785, ...,   200000,    5,
  424],
[   143,   291,   291, ...,    7000,    7,
 19000],
[    43,    16,    86, ...,   1100,    6,
 456]])

```

In [40]:

```
len(X[0])
```

Out[40]:

```
10
```

In [47]:

```
y[np.newaxis].T
```

Out[47]:

```

array([[8],
       [6],
       [7],
       ...,
       [5],
       [7],
       [6]])

```

In [44]:

```
datadf
```

```

def _make_in_format(): datadf = pd.read_csv('movie_metadata_filtered_aftercsv.csv') #separate classes
and stuffs y = np.array(datadf['imdb_score']) datadf = datadf.drop(datadf.columns[[8,0]],axis=1)
#normalize datadf = (datadf-datadf.mean())/(datadf.max()-datadf.min()) X = np.array(datadf) return
X,y

```

In [50]:

```

from sklearn.cross_validation import train_test_split X,y = _make_in_format()
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=0)

```

```
X_train
```

Out[51]:

```

array([[ 1.02685224e-01, -1.75031825e-02, -1.80870525e-02, ...,
        -3.45409216e-03,  1.57035176e-02,  1.54215703e-02],
       [-1.76871426e-01, -4.22444643e-02, -1.80870525e-02, ...,
        -3.24943414e-03, -3.59296482e-01, -3.79945147e-02],
       [-5.74132980e-02, -3.95485777e-02, -2.29563395e-02, ...,
        -3.02358650e-04,  1.57035176e-02, -7.50132858e-03],
       ...,
       [-1.41157140e-01, -4.20270541e-02, -2.46678977e-02, ...,
        -3.43771952e-03,  1.40703518e-01, -3.77538243e-02],
       [-1.32536451e-01, -3.58091221e-02,  6.65293084e-02, ...,
        -5.47948275e-04,  1.57035176e-02, -3.80117069e-02],

```

```
[ -3.22610590e-03, -4.12443774e-02, -2.21101759e-02, ...,  
  -2.02148602e-03,  1.57035176e-02, -3.61578175e-02]])
```

In [52]:

y_train

Out[52]:

```
array([ 0.01570352, -0.35929648,  0.01570352, ...,  0.14070352,  
        0.01570352,  0.01570352])
```

In []:

Conclusion

The [Internet Movie Database \(IMDb\)](#) stands as a remarkable testament to the power of information and community within the realm of entertainment. From its humble origins as a personal project to becoming a global phenomenon, IMDb has redefined how we explore, appreciate, and engage with movies and TV shows.

IMDb's influence on the entertainment industry is undeniable. It serves as an essential resource for filmmakers, actors, and industry professionals, offering a comprehensive platform for networking, research, and project exploration. The IMDbPro subscription further caters to the needs of those working behind the scenes, providing tools to navigate the intricate web of the industry.