IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Machine Learning-Based System for Movie Recommendations

Ms. R.VARSHA KRISHNA #1, Mr. K. CHINNANAGARAJU #2 , Mr. D.D.D.SURIBABU #3

#1M.Tech Student, #2 Associate Professor, #3 Associate Professor & HOD
Department of Computer Science & Engineering,
International School of Technology and Sciences (Women),
(Affiliated to JNTUK),East Gonagudem, Rajanagaram, Rajamahendravaram,
East Godavari District, Andhra Pradesh, India– 533294.

ABSTRACT

The movies in this recommendation are chosen using a content-based recommendation system. We have a dataset of the movies in the project. There are director, keyword, genres, cast, etc. in that dataset. A content-based recommendation engine will consider how much the user enjoyed the film. The next step is to analyse the input movie's content (a dataset), including its genres, director, keywords, and cast, to identify additional films with comparable results. It then evaluates comparable films according to its similarity score and suggests the film to consumers.

KEYWORDS:

Movie Recommendation, Similarity Score, Movies Dataset, Content-Based Search.

1. INTRODUCTION

Every time you open up YouTube just to figure out the solution to your problem or just get the latest news, you end up spending more time. A similar thing happens when you decided on binging through a single movie/series from an OTT you end up watching more than what you had in your mind. Ever wondered how they were able to do such a thing? Most of the OTT platforms depend on their movie recommendation system.A movie recommendation system is a fancy way to describe a process that tries to predict your preferred items based on your or people similar to you. From a user's perspective, they are catered to fulfil the user's needs in the shortest time possible. For example, the type of content you watch on Netflix or Hulu. A person who likes to watch only Korean drama will see titles related to that only but a person who likes to watch Action-based titles will see that on their home screen.From an organization's perspective, they want to keep the user as long as possible on the platform so that it will generate the most possible profit for them. With better recommendations, it creates positive feedback from the user as well.

In the era of information overload, it is very difficult for users to get information that they are really interested in. And for the content provider, it is also very hard for them to make their content stand out from the crowd. That is why many researchers and companies develop Recommender System to solve the contradiction. The mission of Recommender System is to connect users and information, which in one way helps users to find information valuable to them and in another way push the information to specific users. This is the win-win situation for both customers and content providers.

The evolution of technology brings us many advanced platforms such as Machine Learning, Deep Learning, Data Mining, the Internet of Things (IoT), etc. To satisfy the need of society, almost in each work, we use this technology. It has many real-life applications such as PowerShell [1], TP [2-4], IoT [5-12], Cloud Computing [13], Artificial Intelligence [14], Uncertainty [15-17], virtualization Environment [18], SPP [19-26], and so on. IT is the mode to store, fetch, communicate and utilize the information. So, all the organizations, industries and also every individual are using computer systems to preserve and share the information. As we probably are aware, the world is becoming quicker and everybody is moving towards accomplishing their objectives. Individuals need more time to go to the market and purchase things, not simply that, they don't have the opportunity to pick between things.

2. EXISTING SYSTEM AND ITS LIMITATIONS

In the existing system there was no proper method to examine the set of reviews given for some movies and identify the best movie and worst movie and recommend to the users who wish to view the movie..There is no proper recommendation system in the existing system which can able to automatically recommend the set of movies based on keyword. All the existing approaches try to verify the data manually and they try to recommend the movies based on the keywords which are present on the dataset. Hence it is very time complexity to process the task.

LIMITATION OF PRIMITIVE SYSTEM

1. More Time Delay in finding the best possibility of movies.
2. There is no accurate prediction.
3. This is not efficient method to recommend the movies based on raw dataset
4. All the primitive methods use manual approach for movie recommendation.
5. There are no automatic recommendation techniques in existing system.

3. PROPOSED SYSTEM AND ITS ADVANTAGES

Machine learning approach is applied which employs The proposed system is very accurate for recommendation of movies for the end users based on the reviews which are collected from several users. In this proposed system we try to apply ML approach as recommendation model to recommend the movies based on user search keyword. Also the proposed system is very good in finding the recommendation of movies by using meta data recommendation model which can give accurate recommendation of result. Hence by using this proposed movie recommendation using ML approach we can get following advantages like:

ADVANTAGES OF THE PROPOSED SYSTEM

1) By using these two recommendation models such as item based recommendation and content based.

2) We finally get accurate recommendation based on text reviews given by the user.

3) This approach reduces lot of time delay in verifying the raw tweets provided on movies.

4) Here we used meta data recommendation which is very accurate in generating recommendation for the movies.

4. IMPLEMENTATION PHASE

Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a number of modules and then coded for deployment. The front end of the application takes Google Collaboratory. Here we are using Python as Programming Language to implement the current application. The application is divided mainly into following 5 modules. They are as follows:

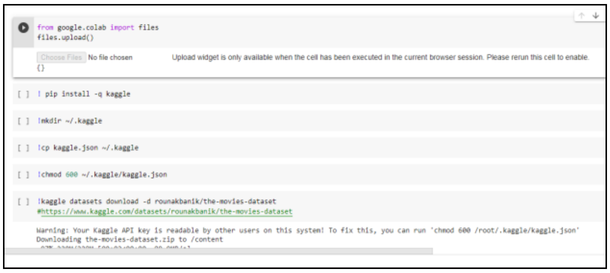
1. Import Necessary Libraries
2. Load Dataset Module
3. Data Pre-Processing
4. Train the Model Using Several recommendation models
5. Find the Performance of movie recommendation

1) IMPORT LIBRARIES MODULE

In this module initially we need to import all the necessary libraries which are required for building the model. Here we try to use all the libraries which are used to convert the data into meaningful manner. Here the data is divided into numerical values which are easily identified by the system, hence we try to import numpy module and for plotting the data in graphs and charts we used matplot library.

2) LOAD DATASET MODULE

In this module the we try to load the dataset which is downloaded or collected from google repository. The description of dataset is as follows:



```
from google.colab import files
files.upload()
```

Choose Files No file chosen   Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

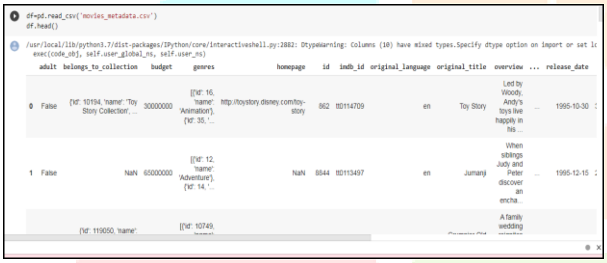
```
{}
```

```
[ ] ! pip install -q kaggle

[ ] !mkdir ~/.kaggle

[ ] !cp kaggle.json ~/.kaggle

[ ] !chmod 600 ~/.kaggle/kaggle.json

[ ] !kaggle datasets download -d rounakbanik/the-movies-dataset
    #https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset
```

```
Warning: Your Kaggle API key is readable by other users on this system! To fix this, you can run 'chmod 600 /root/.kaggle/kaggle.json'
Downloading the-movies-dataset.zip to /content
```

3) DATA PRE-PROCESSING MODULE

Here in this section we try to pre-process the input dataset and find out if there are any missing values or in-complete data present in the dataset. If there is any such data present in the dataset, the application will ignore those values and load only valid rows which have all the valid inputs.



```
df=pd.read_csv('movies_metadata.csv')
df.head()
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2882: DtypeWarning: Columns (10) have mixed types.Specify dtype option on import or set lc
  exec(code_obj, self.user_global_ns, self.user_ns)
```

	adult	belongs_to_collection	budget	genres	homepage	id	imdb_id	original_language	original_title	overview	...	release_date
0	False	{'id': 10194, 'name': 'Toy Story Collection', ...	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...	http://toystory.disney.com/toy-story	862	tt0114709	en	Toy Story	Led by Woody, Andy's toys live happily in his ...	...	1995-10-30
1	False	NaN	65000000	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...	NaN	8844	tt0113497	en	Jumanji	When siblings Judy and Peter discover an encha...	...	1995-12-15
		{'id': 119050, 'name':		[{'id': 10749,						A family wedding		

4) TRAIN THE MODEL

Here we try to train the current model on given dataset using several recommendation models algorithms and then try to find out which algorithms suits best in order to identify and classify the input dataset accurately and efficiently. Here we try to use following algorithms on input dataset such as:

1. Item based
2. Content Based

5) COMPARITIVE ANALYSIS

Here in this module we try to compare each and every classification algorithm on given input dataset and then try to find out which one suits best for finding the accurate results. Finally we try to take one algorithm for recommendation of movies.

5. EXPERIMENTAL RESULTS

In this section we try to design our current model using Python as programming language and we used Google Collab as working environment for executing the application. Now we can check the performance of our proposed application as follows:

IMPORT LIBRARIES

```
from google.colab import files
files.upload()
```

Choose Files No file chosen   Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

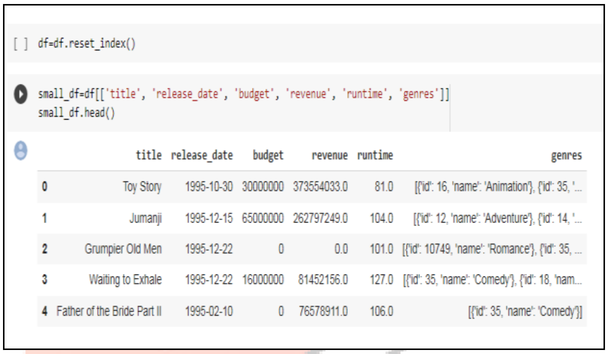
```
{}
```

```
[ ] ! pip install -q kaggle

[ ] !mkdir ~/.kaggle

[ ] !cp kaggle.json ~/.kaggle

[ ] !chmod 600 ~/.kaggle/kaggle.json

[ ] !kaggle datasets download -d rounakbanik/the-movies-dataset
    #https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset
```

```
Warning: Your Kaggle API key is readable by other users on this system! To fix this, you can run 'chmod 600 /root/.kaggle/kaggle.json'
Downloading the-movies-dataset.zip to /content
```

The above window clearly represents the list of several modules used in our application.

PRE-PROCESS THE DATA



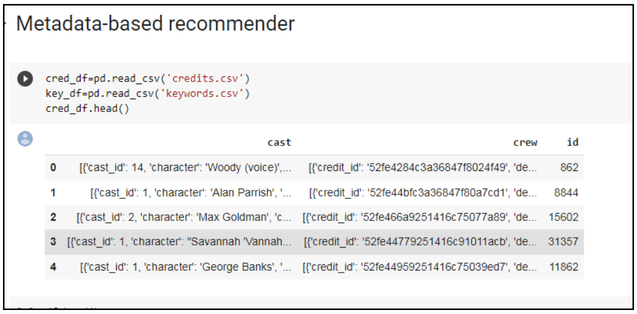
```
[ ] df=df.reset_index()
```

```
small_df=df[['title', 'release_date', 'budget', 'revenue', 'runtime', 'genres']]
small_df.head()
```

	title	release_date	budget	revenue	runtime	genres
0	Toy Story	1995-10-30	30000000	373554033.0	81.0	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...
1	Jumanji	1995-12-15	65000000	262797249.0	104.0	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...
2	Grumpier Old Men	1995-12-22	0	0.0	101.0	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...
3	Waiting to Exhale	1995-12-22	16000000	81452156.0	127.0	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...
4	Father of the Bride Part II	1995-02-10	0	76578911.0	106.0	[{'id': 35, 'name': 'Comedy'}]

From the above window we can see DATA is pre-processed and we converted dataset into test and train.

META DATA RECOMMENDER



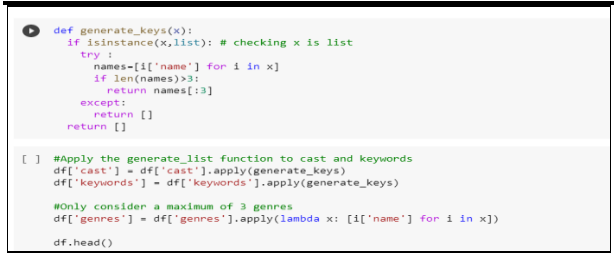
Metadata-based recommender

```
cred_df=pd.read_csv('credits.csv')
key_df=pd.read_csv('keywords.csv')
cred_df.head()
```

	cast	crew	id
0	[{'cast_id': 14, 'character': 'Woody (voice)',...	[{'credit_id': '52fe4284c3a36847f8024f49', 'de...	862
1	[{'cast_id': 1, 'character': 'Alan Parrish', '...	[{'credit_id': '52fe44bfc3a36847f80a7cd1', 'de...	8844
2	[{'cast_id': 2, 'character': 'Max Goldman', 'c...	[{'credit_id': '52fe466a9251416c75077a89', 'de...	15602
3	[{'cast_id': 1, 'character': "Savannah 'Vannah...	[{'credit_id': '52fe44779251416c91011acb', 'de...	31357
4	[{'cast_id': 1, 'character': 'George Banks', '...	[{'credit_id': '52fe44959251416c75039ed7', 'de...	11862

From the above window we can clearly see the meta recommender.

GENERATE CLASSIFICATION

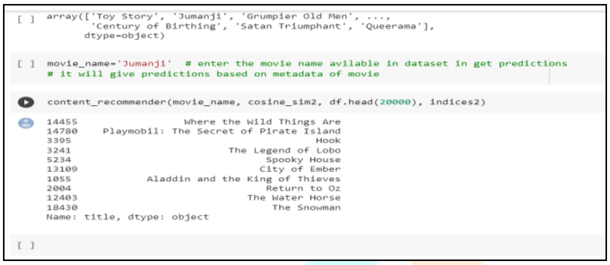


```
def generate_keys(x):
  if isinstance(x,list): # checking x is list
    try :
      names=[i['name'] for i in x]
      if len(names)>3:
        return names[:3]
    except:
      return []
  return []
```

```
[ ]  #Apply the generate_list function to cast and keywords
     df['cast'] = df['cast'].apply(generate_keys)
     df['keywords'] = df['keywords'].apply(generate_keys)

     #Only consider a maximum of 3 genres
     df['genres'] = df['genres'].apply(lambda x: [i['name'] for i in x])

     df.head()
```

From the above window we can clearly see the generate classification.

MOVIE RECOMMENDATION



```
[ ]  array(['Toy Story', 'Jumanji', 'Grumpier Old Men', ...,
            'Century of Birthing', 'Satan Triumphant', 'Queerama'],
           dtype=object)

[ ]  movie_name='Jumanji'  # enter the movie name avilable in dataset in get predictions
     # it will give predictions based on metadata of movie

     content_recommender(movie_name, cosine_sim2, df.head(20000), indices2)

     14455                 Where the Wild Things Are
     14780    Playmobil: The Secret of Pirate Island
     3395                                       Hook
     3241                         The Legend of Lobo
     5234                               Spooky House
     13109                             City of Ember
     1055             Aladdin and the King of Thieves
     2004                               Return to Oz
     12403                           The Water Horse
     18430                               The Snowman
     Name: title, dtype: object

[ ]
```

From the above window we can clearly see the movie is recommended.

6. CONCLUSION

Recommendation systems have become an important part of everyone's lives. With the enormous number of movies releasing worldwide every year, people often miss out on some amazing work of arts due to the lack of correct suggestion. Putting machine learning based Recommendation systems into work is thus very important to get the right recommendations. We saw content-based recommendation systems that although may not seem very effective on its own, but when combined with collaborative techniques can solve the cold start problems that collaborative filtering methods face when run independently. Similarly such systems can be improved further by applying neural network embeddings to uplift the quality of recommendations and make them more user personalized. Thus we conclude that studying various approaches towards recommendation engine is vital to come up with a hybrid engine that overcomes the shortcomings of these independent approaches and multiplies their benefits. Where independent approaches towards a movie recommendation system may have shortcomings, when combined the right way they will help users get the accurate recommendations for movies.

7. REFERENCES

[1] Mohapatra, H., Panda, S., Rath, A., Edalatpanah, S., & Kumar, R. (2020). A tutorial on powershell pipeline and its loopholes. International journal of emerging trends in engineering research, 8(4), 975-982.

[2] Kumar, R., Edalatpanah, S. A., Jha, S., & Singh, R. (2019). A Pythagorean fuzzy approach to the transportation problem. Complex & intelligent systems, 5(2), 255-263.

[3] Smarandache, F., & Broumi, S. (Eds.). (2019). Neutrosophic graph theory and algorithms. Engineering Science Reference. Kumar, R., Edalatpanah, S. A., Jha, S., & Singh, R. (2019).

[4]A Pythagorean fuzzy approach to the transportation problem. Complex & intelligent systems, 5(2), 255-263.

[5] Mohapatra, H. (2009). HCR using neural network (Doctoral dissertation, Biju Patnaik University of Technology). Retrieved from https://www.academia.edu/39142624/HCR_USING_NEURAL_NETWORK

[6] Mohapatra, H., & Rath, A. K. (2019). Detection and avoidance of water loss through municipality taps in India by using smart taps and ICT. IET wireless sensor systems, 9(6), 447-457.

[7] Mohapatra, H., & Rath, A. K. (2019). Fault tolerance in WSN through PE-LEACH protocol. IET wireless sensor systems, 9(6), 358-365.

[8] Mohapatra, H., Debnath, S., & Rath, A. K. (2019). Energy management in wireless sensor network through EB-LEACH (No. 1192).

[9] Easy Chair. Nirgude, V., Mahapatra, H., & Shivarkar, S. (2017). Face recognition system using principal component analysis & linear discriminant analysis method simultaneously with 3d morphable model and neural network BPNN method. Global journal of advanced engineering technologies and sciences, 4(1), 1-6.

[10] Panda, M., Pradhan, P., Mohapatra, H., & Barpanda, N. K. (2019). Fault tolerant routing in Heterogeneous environment. International journal of scientific & technology research, 8(8), 1009- 1013.

[11] Mohapatra, H., Rath, A. K. (2020). Fault-tolerant mechanism for wireless sensor network. IET Wireless sensor systems, 10(1), 23-30.

[12] Swain, D., Ramkrishna, G., Mahapatra, H., Patr, P., & Dhandrao, P. M. (2013). A novel sorting technique to sort elements in ascending order. International journal of engineering and advanced technology, 3(1), 212-126.

[13] Xu, X. (2012). From cloud computing to cloud manufacturing. Robotics and computer-integrated manufacturing, 28(1), 75-86. Furtado & Singh / Int. J. Re