A comparative analysis of machine learning approaches for movie success prediction

Ankit[1], Gautam Arora[2]

[1]*Student, Department of Computer Science And Engineering, SDDIET, Barwala, Haryana*
[2]*Assistant Professor, Department of Computer Science And Engineering, SDDIET, Barwala, Haryana*

Abstract - *The success of a movie is crucial for hundreds of people who labour behind the scenes as well as the movie's producers. They rely for their subsistence money generated by the film. The precise foreseeing of a. It's difficult to predict if a movie will be successful or unsuccessful because it has a lot of unknown parameters. In light of this, the machine learning (ML) use in determining if movie will succeed or fail may significantly lower the financial burden shared by all parties. The emphasis of this article is on creating a program that can assist in anticipating the movie's early success will encourage investors to invest analysis is done on some of the patterns from the movie the IMDb collection. Using data gathered from several sources and the system uses a variety of machine learning methods estimates a film's likelihood of success based on its success by looking at historical data from places like IMDb, Reputable Tomato. Experimental findings show that the scores are really outstanding throughout the testing stage. Additionally paper ends by identifying the top actors or actresses in to ensure that the film makes the most money possible. This investigation highlights the value of prediction in the professional realm. Since only these projections serve as the foundation for all capital investments.*

Key Words—- Movie; Machine Learning; Prediction, Hit; Flop; SVM; k-NN; GNB

1. INTRODUCTION

Modern film business is tremendously lucrative, creating a huge area so as to invest. Film investors incur several threats, thus their choice should be extremely carefully considered precise else, they risk incurring enormous debt. Numerous data are accessible from a variety of sources [1]. This planned construction would benefit both the investors and the general public, who may choose whether to view this film or not. The criteria for the success of a film vary depending on the genre. A film's worldwide box office performance, and some Movies may not be as effective at generating income but they can praised for its excellent reviews, ratings, and popularity [2,3]. Various ML methods are used in this paper for predictions. Support Vector Machines (SVM) are them. Both k-Nearest Neighbor (kNN) and Gaussian Naive Bayes (GNB). These algorithms combine the data from actor(s), genre, director, and budget of the film. From the 5000-movie IMDB dataset, movies that have previously produced hits are used to predict future box office success. In this manner, it aids filmmakers in selecting the ideal cast of actors and actresses for any genre. The outcome of mentioned model is either a hit or a flop. This process verifies each input combination before determining whether to label the film a success or a failure. Movie title, director's name, actor's name, and actress name are a few of the characteristics that are entered. There were initially many misconceptions about the traits to choose. We employed feature selection, also known as variable selection, to overcome this problem. In this method, every subset of the variables were selected. It was the most important and essential element. The model has a very high possibility of failing if the wrong qualities were picked.

There are 5 parts left in the paper. Part II discusses the pertinent work. The suggested work is shown in part III. The obtained result findings are shown in part IV. The report concludes with a discussion on future research.

2. RELATED WORK

There was a lot of study done on this subject in the past. Some earlier efforts used IMDB data to determine their success. Depending on how much money a movie makes, some study divides the work in essentially two categories: hit or failure. We cannot claim that a film's success is only based on its box office performance. The actors, actresses, director, shooting location, screenwriter, music director, etc, all have a role in a movie's success.

Some academics calculated the success using historical data. For testing purposes, several studies have made extensive use of NLP systems for collecting movie reviews. Many individuals left reviews for the movie even though they had not seen it on all the screens. Because audience reviews might be skewed by an actor or actress's fan base.

In [4], the author created a decision-making system to forecast the box office success utilizing machine learning methods, data mining, and social networks. Their analysis revealed dynamic network connectivity. Their study was mostly based on the elements of who the main actor or actress in the film is, what the film's overall budget is, when it will be released, and how much money the film will ultimately make. They divided the success of movies into three categories: audience, release, and film. Their primary method of forecasting was based on the idea that if the audience is more upbeat, enthusiastic, or happy, the likelihood that the film would be profitable will increase. Similar to this, if a film is more negative and draws fewer

viewers, its revenue will suffer. They gathered the data from a variety of websites, including Facebook, Twitter, blogs, and YouTube comments. They obtained the information from Box-office mojo and IMDB. Their primary concentration was on films that were released in the United States; they did not include any films from other nations.

Authors attempted to forecast movie sales using buzz analysis in [5]. They gathered Twitter data for research on public relations.

The basic goal of hype analysis is to forecast a film's commercial success based on its first week's earnings and the buzz it generated before its release. Using a web crawler, they discovered the amount of tweets about a movie that were accessible. These tweets are being gathered day by day. There are three steps in determining a movie's success, who can count the most tweets per second. The number of unique users who posted the tweet was the second consideration, and determining the message's reach was the third. Their forecast also took into account how many screens the image was shown on. The average ticket price was also taken into account.

The failure to evaluate the tweet and determine whether it was favorable or negative was one flaw in this effort.

The tweets were only counted. The income produced by the image was predicted using a neural network.

The author of [6] attempted to take news analysis into account while making success predictions. The likelihood of the movie succeeding is increased if the news is good. Both quantitative and qualitative news should be presented. Regression and k-NN are two other algorithms she employed for her predictions.

However, one of its drawbacks was that she only watched expensive movies. This model was unsuccessful because it is possible for a movie to come out without any news around it, making it impossible for the algorithm to make any predictions. The IMDb database was used by writers. The data was quite noisy and had not been cleansed. As a result, used their methods to replace the empty information.

3. PROPOSED WORK

The suggested study to forecast movie success using ML algorithms is described in this part. The Kaggle website's [6] dataset is used. 4000 movies are represented by 11 attributes in the data. There are hundreds of performers and actresses along with 1819 distinct directors. The dataset's attribute description is shown in TABLE I.

TABLE I. ATTRIBUTE DESCRIPTION [6]

Attribute Name	Description
Movie_Title	The movie's label
Director_Name	Name of the film's director
Actor_1	Primary actor in the film
Actor_2	Supporting Actor in the Film
Actor_3	Supporting Actor in the Film
Genres	What Sort of Film?
IMDb_Rating	IMDb reviews for the film
Budget	The total funds utilized for the film
Gross	Revenue for the film
Profit_Percentage	Movie's Profit Margin
Hit/Flop	Whether the film is Hit or Flop

Six main parts of the approach are listed below.

Data Collection: Data gathering is a crucial component of every machine learning (ML) project. The IMDb or rotten tomatoes dataset, which was often used for testing, was used for this study.

Filling in Missing Values: Some rows may contain some values that are missing. Therefore, the missing values required to be replaced. Mean/median imputation is an option. With this technique, the mean or median of the feature is used to fill missing row information.

Data purifying: We had to get rid of the excess row since it could have been superfluous.

Inappropriate findings: Inappropriate findings are ones that are not necessary for analysis, or irrelevant observations. That won't make the observation any easier.

Data Structure: After eliminating unnecessary observations, fix the data structure. The proper organized order is now very necessary. As a result, algorithms can compute things more quickly.

Information analysis and forecasting: Actor, Director, IMDb_Rating, Genres, and Budget are the characteristics needed for projections. The data has now been divided into training and testing datasets.

3.1) Architectural Design

A prediction model's architecture is shown in Fig. 1. The dataset was gathered from an additional source. Various characteristics included in the dataset. The dataset can

include a number of errors, such as a series of values that are absent, which need to be fixed. We can impute using the mean, median, and mode. In this approach, the feature average is used to fill in empty values. This input data can even include unimportant data that have to be processed in accordance with the requirements of mentioned algorithms. We now have the cleaned dataset. We used a variety of machine learning algorithms, on this cleaned dataset to see which performed best. The final algorithm's output will provide us with patterns. We will receive results if we investigate these patterns.

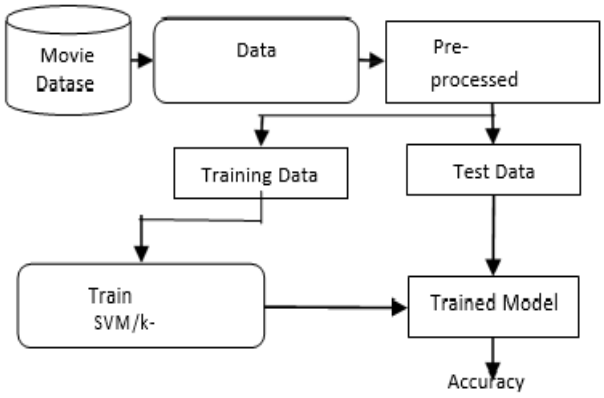

Fig. 1. Architectural design

3.2) Data Visualization

A crucial component of our effort is data visualization. Grossly were all visualized here. We were able to identify the best actors, directors, and popular genres, thanks to these visualizations.

i) IMDB Scores Vs Gross Graph

In Fig.2. We evaluated against the total value at various IMDB score values. Therefore, we used gross as the y-axis and IMDB_Rating as the x-axis in this graph. Here, we utilized the graph's scatter function to scatter the data points according to the IMDB rating.

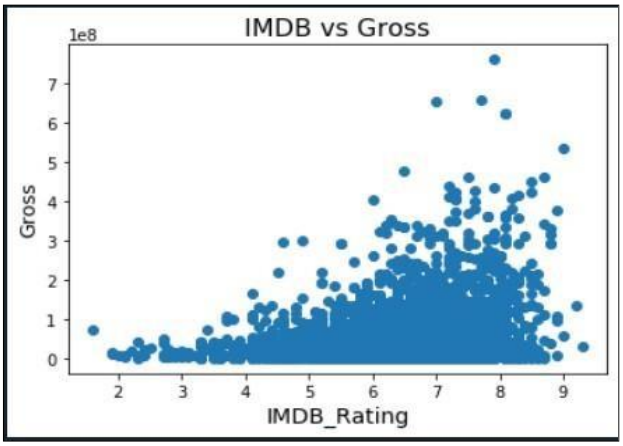

Fig. 2. IMDb Vs. Gross Graph

We can see that distinct points of gross are noted at different IMDB_ratings in the graph shown in Fig. 2. From this graph, it is clear that a movie has a greater chance of being successful if it received a rating of higher than 8. Here, with around an 8.2 rating, we can see the movie with the largest box office haul. The majority of movies with an IMDb rating below 8 had weak box office performance.

Primary Actor Mean Gross

Fig.3 depicts graph where we may locate the 20 best from starting.

ii). Actor Mean Gross

The actors' mean gross is shown in descending order in Fig. 3. Here, we can see that Rupert has the largest mean gross of any first actor. (4.3 * 108) is Rupert's mean gross.

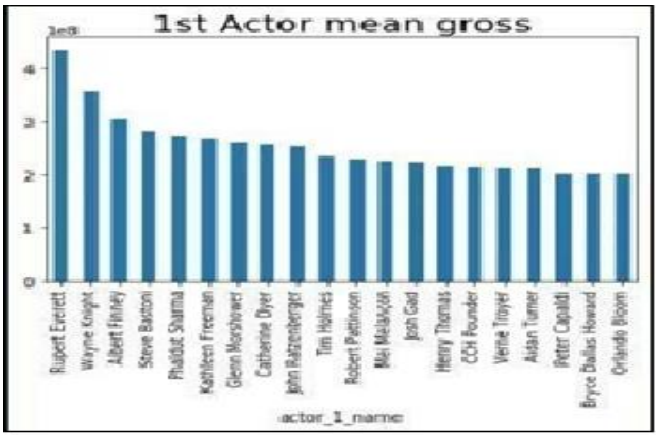

Fig. 3. Primary Actor Mean Gross Graph

The actor with the lowest mean gross may be identified by printing the bar graph in increasing order of the actors' mean grosses.

iii). Genres Mean Gross

We identify the genres that the audience will like the most in the bar graph (Fig. 4). The top 12 genre categories are shown below. There are several genre subtypes, including Family. Science fiction, action, animation, and romantic drama.

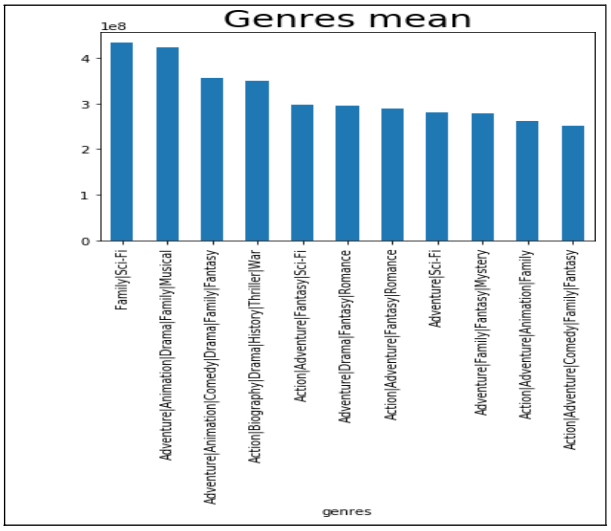

Fig. 4. Genres mean gross code

It is clear from Fig. 4 that the majority of viewers like films Sci-Fi/Family, concluded that these films had a good chance of becoming successful as a result.

3.3) Algorithm implementation

Different algorithms, including k-NN, SVM, and GNB classifier, were employed in the algorithm implementation section. The implementation of the method is briefly discussed in this section.

i) k-NN:

For forecasting the k-NN is applied to the new value in either class. This method may be used across a range of distances. We used the Euclidean distance as the closeness metric in this endeavor. Figure 5 shows how this approach classifies a certain node (information entry or instance) [7].

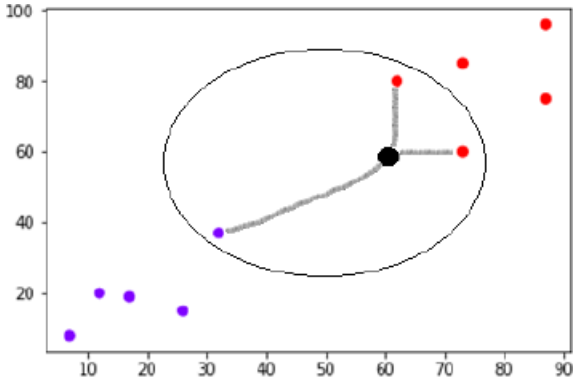

Fig. 5. k-NN Classification

These above mentioned dots of blue color as shown in Fig. 5 are records that belong to first class, while the red dots are records that belong to a different class, let's say second class. Consider that a prediction is required for the data point in black. The fresh anticipated information, further separated as belonging to the class of the red point if it is near to red dots; otherwise, should be identified as relating to the blue point class. In our work [8], we use Euclidean distance to quantify this proximity.

ii). SVM:

We want to optimize the space between the planes in SVM. In SVM, a straight line is used to divide the planes. The fundamental idea of categorization as it occurs in SVM is shown in Fig. 6 [9].

The blue dots in Fig. 7 indicate data instances that fall under Class 1 (C1), whereas the red dots fall under Class 2 (C2). Depending on the maximum margin and hyperplane concepts, fresh information shall then be classified as fitting to C1 or C2. New data will be classed as belonging to C1 if it is near to red-valued information, otherwise as C2 [10].

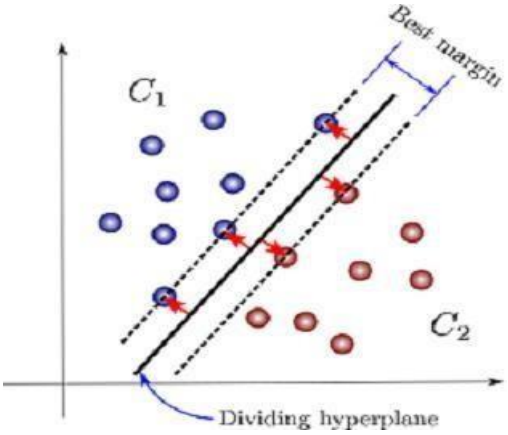

Fig. 6. SVM classification

iii). GNB:

The foundation of the Bayes theorem is from GNB classifier. It is founded on the concept of conditional probability [11]. This classifier of GNB is shown in Algorithm 1 for predicting film triumph.

Algorithm-1: Gaussian Naïve Bayes Classifier for MovieSuccess Prediction

Input: *Information required for film*

Output: *Hit or Flop*

1: It begins out with collection of input samples.

.2: The input data is divided into m parts.

```
copy_inputSet = list(input)

sizeofFold = int(length(dataset) / m_part)while
length(part) < sizeofFold:

    index =

     randomrange(len(copy_inputSet))
    part.append(copy_inputSet.pop(index))
    data_split.append(part)
```

3: Diving the set into training and testing data.

4: Calculate the probailty density function

```
exponent=math.exp(-(math.pow(x-mean,2)/( 2
 math.pow(stdev,2)) 1 / (math.sqrt(2 * pi)

 *standeviat) * expo.
```

5: With the use of test data, determine the prediction using training data as an input.

6:Need to evaluate accuracy market:

```
Here the correctness of the small parts will be
determined i.e no. of correct prediction Correct /
float(length(actual)) * 100.0
```

7:Repeat

8:Find the precision of each *part*

9:Calculate the average accuracy

```
add(navieAccuracy) / length(navieAccuracy)
```

The above mentioned algorithm, that is Algorithm-1 shows that k-fold cross validation (kcv) is being used. In kcv, accuracy is calculated for each fold once the data is separated into k folds. The average of all the k folds' accuracy is then calculated. This provides the ML model's ultimate accuracy. An accuracy of 85.8% was attained using the GNB. Recall is 82.2% and precision is 86.4%.

Some of the modifications are:

1. Statistics using the one-way analysis of variance utilized to determine the importance of data in at least two categories, statistically speaking. Because an independent ANOVA considers singular remedy applied over many tiers, we chose it over two-way ANOVA. When the data is close to the mean, it functions well. As indicated in Equation, the analysis uses below mentioned formula, where MS is mean square.

1. F = (MS within) / (MS Between) (1)

2. Utilize Probabilities – Probabilities are often quite modest when they are discovered. The quantity decreases by a relatively modest amount when joint probability is discovered. Finding the outcome with such a little number is challenging; hence, we employed the probability logarithm to prevent this.

3. Less Data - Because naive Bayes requires relatively little data. The difference is fairly little for large amounts of data, but the algorithm's processing time is crucial. Therefore, relatively little data is needed to function successfully.

4. EXPERIMENTAL DESIGN AND OUTCOMES

We go through the setup, snapshots, and outcomes of this effort in this part. It contains a thorough analysis of every significant test's outcomes that was performed. The studies were run on a Windows 10 computer with two TB hard drive and 6 GB of RAM. Spyder, a version of Python 3.4, was the program utilized.

The input screen where users submit their inputs is shown in Fig. 7.

All of the factors needed to determine whether a movie will be a success or a failure are shown in Fig. 7. The output from inputting attributes value is shown in Fig. 8. After receiving the user's input, we employed our algorithm, which uses these inputs as parameters to determine if the movie is a hit or a failure.

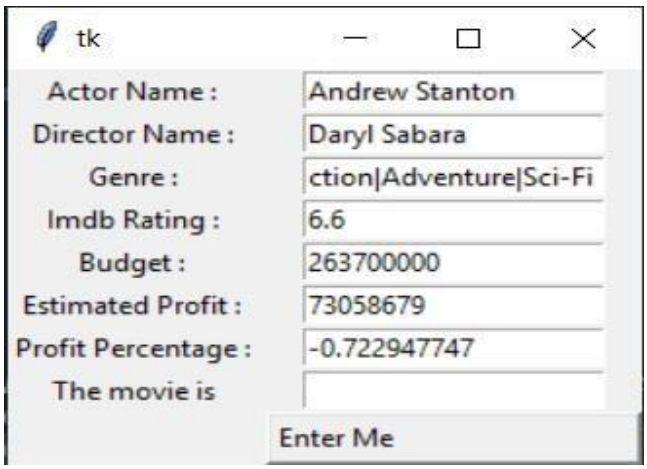

Fig.7 Passing required attributes as inputs

The ultimate value of the prediction must equal 1, in which case the file is said to be a success; otherwise, it is predicted to be a failure. Using the predict technique, the prediction's value is determined.

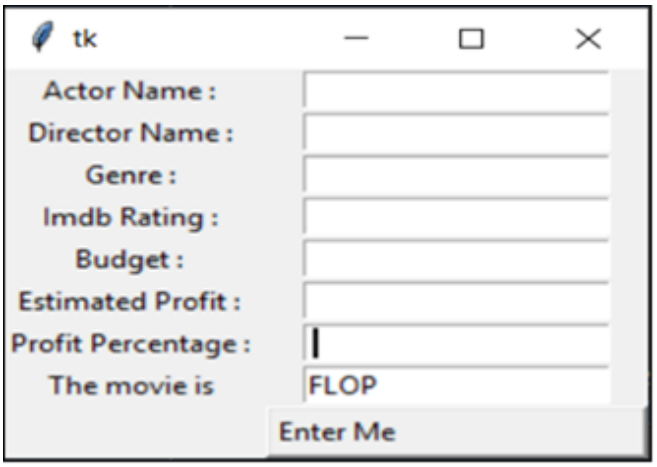

Fig. 8 Forecast of the working application model

In Fig. 8, the projected output for the specified input is FLOP. Here, the prediction method's final value is not equal to 1, which is why the film's outcome is displayed above as a failure.

Fig. 9 shows how the classifiers compare in terms of the performance standards, Accuracy, Precision, and Recall.

As shown in Fig. 9, GNB outscored SVM, k-NN, and other algorithms in terms of performance. The performance metrics for each of the aforementioned algorithms are shown in Table-II.

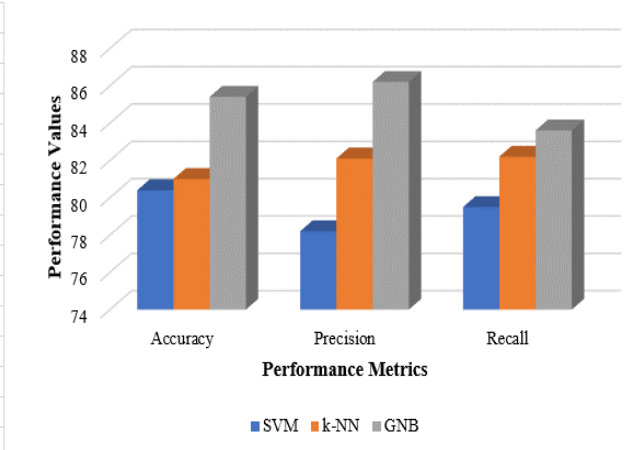

Fig. 9. Comparison of algorithms for predicting movie success.

Table II. PERFORMANCE VALUES.

Algorithms	Accuracy	Precision	Recall
SVM	80.4	78.2	79.5
k-NN	81	82.1	82.2
GNB	85.4	86.2	83.6

As a result, we may conclude that the GNB method performed best for this dataset, with accuracy increases of 5% over SVM and 4.4% over k-NN. Additionally, the GNB algorithm improved accuracy & recall by 8%, 4.1%, and 1.4% over SVM, respectively. GNB's superior performance may be explained by the fact that it works better with less datasets than SVM and k-NN, which need big datasets for training.

5. CONCLUSION AND FUTURE SCOPE

By highlighting the key components of each portion, this section highlights the whole work. In this study, the result of a film was predicted to be hit or failure. The different attributes which are mentioned above are the input criteria that are taken into account for the forecast. Cleaning and integrating the extra data are necessary. We translated the majority of the data to numerical form since text-based data is difficult to utilize as input. It is anticipated in this work that the film's production costs are offered. If not, it would be exceedingly challenging to get the desired outcomes. We used the three well-known machine learning (ML) algorithms k-NN, SVM, and GNB to predict whether the film would be a success or a failure. Results showed that compared to SVM and k-NN algorithms, the GNB carried out precisely improvements of 5% and 4.4%, respectively. In the future, the work may be expanded to incorporate more vital input factors that affect a film's likelihood of success or failure. To learn more, experiments on big datasets may also be run.

6. REFERENCES

[1] Amit. Kanitkar, "Bollywood Movie Success Prediction using Machine Learning Algorithms," 3rd International Conference on Circuits, Control, Communication and Computing (I4C), Bangalore, India, pp. 1-4, 2018.

[2] Ramesh Dhir, and Raj Kumar, "Movie Success Prediction using Machine Learning Algorithms and their Comparison," First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar,India, pp. 385- 390, 2018.

[3] Jeffrey S. Simonoff and Ilana R. Sparrow, "Predicting Movie Grosses: Winners and Losers, Blockbusters and Sleepers," Chance, vol. 13, no.3, pp. 15-24, 2000.

[4] Mohanbir S. Sawhney and Jehoshua Eliashberg, "A Parsimonious Model for Forecasting Gross Box-Office Revenues of Motion Pictures," vol. 15, no. 2, pp. 113–131, 1996.

[5] https://www.kaggle.com/orgesleka/IMDbmovies, Accessed onlineOctober,2019.

[6] Prashant Rajput, Priyanka Sapkal, and Shefali Sinha, "Box Office Revenue Prediction Using Dual Sentiment Analysis International Journal of Machine Learning and Computing, vol. 7, no. 4, August 2017".

[7] Parimi R., Caragea D, "Pre-release Box-Office Success Prediction for Motion Pictures, In: Perner P. (eds) Machine Learning and Data Mining in Pattern Recognition. MLDM 2013". Lecture Notes in Computer Science, vol. 7988. Springer, Berlin, Heidelberg, 2013.

[8] S. Gopinath, P. K. Chingunta and S. Venkat, "Blog, Advertisement, and Movie Box Office Performance," Management Challenges, vol. 15, no. 12, pp. 2670–2685, 2013.

[9] Muthukumar, Vignesh, and N. Bhalaji. "MOOCVERSITY-Deep Learning Based Dropout Prediction in MOOCs over Weeks." Journal of Soft Computing Paradigm (JSCP), vol.2, no. 3, pp. 140-152, 2020

[10] Raj, Jennifer S, "A comprehensive survey on the computational intelligence techniques and its applications.", Journal of ISMAC, vol. 1, no. 03, pp.147-159, 2019

[11] J. Ahmad, P. Duraisamy, A. Yousef, and B. Buckles, "Movie success prediction using data mining," 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, Delhi, pp. 1-4, 2017

[12] N. Quader, M. O. Gani, D. Chaki, and M. H. Ali, "A machine learning approach to predict movie box-office success," 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, pp. 1-7, 2017.