Multimedia Tools and Applications (2023) 82:9597–9626
https://doi.org/10.1007/s11042-021-11553-0

1222: INTELLIGENT MULTIMEDIA DATA ANALYTICS AND COMPUTING

Predicting attributes based movie success through ensemble machine learning

Vedika Gupta[1] · Nikita Jain[1] · Harshit Garg[1] · Srishti Jhunthra[1] · Senthilkumar Mohan[2] · Abdullah Hisam Omar[3] · Ali Ahmadian[4,5]

Received: 2 February 2021 / Revised: 2 August 2021 / Accepted: 9 September 2021 /
Published online: 6 January 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

The film industry has grown into a multi-billionaire industry in terms of entertainment. The success of the film industry depends on the criteria that how much profit a movie would make which gives the tag of a 'hit' or a 'flop'. Predicting the success is guided by various factors like genre, date of release, actors, net gross and many more. Understanding the stakes involved with a movie release that can affect its success or a failure, before-hand can be a great step towards the expansion of the film industry business. Therefore, this study proposes an ensemble learning strategy as a solution to analyze such understanding where predictions from previously guided attribute calculations can be used to enhance future success/failure accuracy. This study shows various strategies used in the literature to analyze and compare the results obtained. The various machines learning algorithms SVM, KNN, Naive Bayes, Boosting Ensemble Technique, Stacking Ensemble Technique, Voting Ensemble Technique, and MLP Neural Network are applied on the dataset to predict the box office success of a movie. The paper uses various algorithms and their trends in predicting the outcome of a movie and shows that the proposed methodology outperforms the existing studies. The most effective algorithm in the study is Gradient Boosting with a success rate of 84.1297%.

Keywords Boosting ensemble technique · IMDb · KNN · Machine learning · Movie success prediction · MLP-NN · Naive bayes · SVM · Voting ensemble technique

✉ Ali Ahmadian
ali.ahmadian@ukm.edu.my

[1] Department of Computer Science & Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India

[2] School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamilnadu, India

[3] Faculty of Built Environment and Surveying, Universiti Teknologi Malaysia, 81310 Skudai, Johor Bahru, Malaysia

[4] Institute of IR 4.0, The National University of Malaysia, UKM, 43600 Bangi, Selangor, Malaysia

[5] Department of Mathematics, Near East University, Nicosia, TRNC, Mersin 10, Turkey

1 Introduction

Movie industry has been expanding all across the world for a long time. Movies are a source of entertainment for the people who have built interest and desire among them to learn as well as enjoy the source of entertainment. In earlier times, television was the only source where people could enjoy their lives. But as time flew, the movie industry was set up which introduced another platform for people as a source of happiness and entertainment. The movie industry also provides a platform for generating brought jobs, revenue, and infrastructure development of the location. It impacts the economy worldwide, which has increased exponentially over time.

For making a movie, there are various attributes that are taken into consideration like genre, cast, writer, director, producers etc. Movie industry supports and presents every kind of environment from a comedy movie to thriller, inspirational and devotional content. Every year, thousands of movies are launched, and each of them is declared either as a hit or a flop [13]. This hit and flop decide on the net profit which is Net income (movie rights + tickets sold)—Net cost (making cost + promotions). Hence, if the net profit is greater than zero, it's a hit else flop. Internet Movie Database (IMDb) is a platform where the complete database of movies, including all of their attributes is maintained. On this data, data mining techniques could be applied to study the variations and thus, could be utilized for the hit/flop prediction of a movie even before its release [2]. Like IMDb, there is another platform named Box Office India (BOI), where similar data information is available and is restrict to Bollywood movies only. Thus, data used in this study has been extracted from Box Office India (BOI) and Internet Movie Database (IMDb). These two platforms provided all the necessary attributes that are required for the movie hit/flop prediction like genre, budget, gross, cast, directors, and writers.

This manuscript present research on the following groundbreaking question: Can we predict the success or failure of an upcoming or new movie by analyzing the given attributes? In this study, the dataset[1] was created with all the desired attributes. This dataset was extracted and merged from two platforms, IMDb and BOI. All the desired attributes were considered on which various algorithms were implemented. The baseline models such as, support vector machine (SVM), k-nearest neighbors (KNN) etc. were taken on which the dataset was trained and implemented. After ruling out the baseline models, in order to improve the accuracy different permutation and combinations of the baseline models were considered and applied to different ensembles. The ensembles gave a comparatively better results from the baseline models. Along with the combinational implementation of baseline models on the ensembles, some pre-defined ensembles were also tested. After implementing those, the paper concluded that the pre-defined ensembles out performed all the previously obtained results and gave the highest accuracy. Thus, the prepared dataset could give the best accuracy using different methodology such as, the baseline models, ensembles (ref. to Sect. 5). In future, the paper would prefer to get more efficient algorithms with improved accuracy on this dataset.

Therefore, all the research and implementation of our preliminary tests, the most successful and accurate among all of the algorithms are explained in this study. The layout of our study is as follows: Sect. 2 defines motivation and contribution of the paper; Sect. 3 contains the related works; Sect. 4 contains material followed by Sect. 5 describing the methods used in this study. Section 6 explains the result and analysis over the methodologies adopted. Section 7 discusses the conclusion and future scope of the work.

[1] https://github.com/imdb-1951/bollywood_2000-2019.git

Table 1 Existing contributions in movie rating analytics

Authors	Algorithm used	Methodology	Accuracy (%)
Doshi et al. [7]	Regression	Sentimental analysis	80
Quader et al. [24]	MLP	Neural network	89.27
Zhang et al. [31]	K-nearest neighbor	Machine learning model	96.81
Jain [9]	PT-NT ratio	Sentimental analysis	64.4
Dhir et al. [5]	Random Forest	Machine learning model	92.08
Latif et al. [14]	Simple Logistic Algorithm	Machine learning model	84.34
Jernbäcker et al. [10]	SVM	Classifying model	68.8
Verma et al. [27]	Random Forest	Machine learning models	Range (80–90)
Pradeep et al. [23]	J48	Decision Tree algorithms	84.67
Modi et al. [17]	Classifiers	Sentimental analysis	82.99

2 Motivation and contribution outline

The motivation behind the work performed in the paper lies in the need to predict the success/failure rate of a movie before-hand to enhance the growth of the film industry. Predicting the movie rating would help the film industry to target the audience accordingly and manage the cost saving. The contributions of the paper are as follows:

1. The paper proposes various ensembles to predict the success/failure of a movie which has outperformed the existing work.
2. It also presents an analytical view of the attributes that affects the movie rating.
3. The paper also shows a comparison between the existing work presented by various authors. It also emphasizes the flaws that have been included in this study.

3 Related works

In literature, studies have been done on movie prediction using various methodologies. Different studies have been recorded having different accuracies and model implementation techniques. Table 1 shows the different studies done on Movie Success Prediction. Doshi et al. [7] explained the movie prediction using sentimental analysis. They attained an average accuracy of 80%. Quader et al. [24] obtained a highest accuracy of 89.27% using neural network for analyzing movie box office success rate. Here, authors in [31] also studied and implemented various machine learning models and gained a successful accuracy of 96.81% after recording the test analysis gained from the reviews from different resources. In [9], authors have used a PT-NT ratio as their technique and gained an accuracy of which is quite low in comparison with other after using the sentimental analysis methodology. Raj et al. [5] studied on movie prediction and implemented various machine learning models such as random forest, linear regression and attained a highest accuracy of 92.08% from random forest algorithm.

One of the famous studies was done by Latif et al. [14] on movie success rate prediction. They gained the highest accuracy of 84.34% after implementing various machine learning algorithms. Authors in [10] also studied various algorithms to predict the movie success rate and obtained 68.8% accuracy on applying SVM algorithm. Later came the contribution by Verma et al. [27], who implemented various machine learning models. They could justify their accuracy ranging from 80 to 90% with random forest highlighted at the top of all the other

models applied. Authors in [23], also tried implementing some baseline models for the movie success prediction. They came up with different decision tree algorithms and ranked J48 as the highest among all the other algorithms in this category with an accuracy of 84.67%.

Later, authors in [17] decided to work on movie success rate prediction. They started their study and analysis. They used different classifier to study the success rate using sentimental analysis i.e., from reviews and other comments from different sections and gained a round off accuracy of 82.99% as their best score.

Table 1 shows all such studies, their methodology and algorithm used. Existing studies [21] have also made attempts to analyze the sentiment from IMDb reviews. They determined the audience's viewpoints on various aspects of movies [22]. Thus, it concludes that for every research done by the authors. Still, there are a count of research gaps to be filled by considering a greater number of parameters and attributes in order to attain better results.

4 Material

4.1 Ensemble learning algorithms

Ensemble learning algorithms uses a very basic idea. They combine the decision from multiple models to improve the overall performance [6]. It is a well—known technique which is used to improve the performance of a model. Ensembles have various techniques to do this.

In this study, two popular ensemble techniques i.e., max voting, boosting were used. These were the techniques used in this study and based on these techniques various algorithms were studied. The algorithms studied in this study were:

1) Voting classifier
2) Gradient Boosting
3) AdaBoost
4) XGBM (XG Boost)

Algorithm 1: Preprocessing function for the dataset

```
1. function preprocessing(imdb_dataset)
2.       imdb_dataset['Release_month'] = converttomonth(imdb_dataset['Release_Date'])
3.       productive_attributes = ['Release_month', 'IMDb-rating', 'Duration', 'Genre', 'Budget', 'Hit_flop']
4.       imdb_dataset = imdb_dataset[productive_attributes]
5.       X = imdb_dataset['Release_month', 'IMDb-rating', 'Duration', 'Genre', 'Budget']
6.       Y = imdb_dataset['Hit_flop']
7.       function scale_zerotofive(column_name)
8.                return scaling criteria
9.       endfunction
10.      function scale(X)
11.               for each a ∈ X.columns do
12.                        mapping = scale_zerotofive(a)
13.                        X[a] = mapping[X[a]]
14.               endfor
15.               return(X)
16.      endfunction
17.      return X,Y
18. endfunction
```

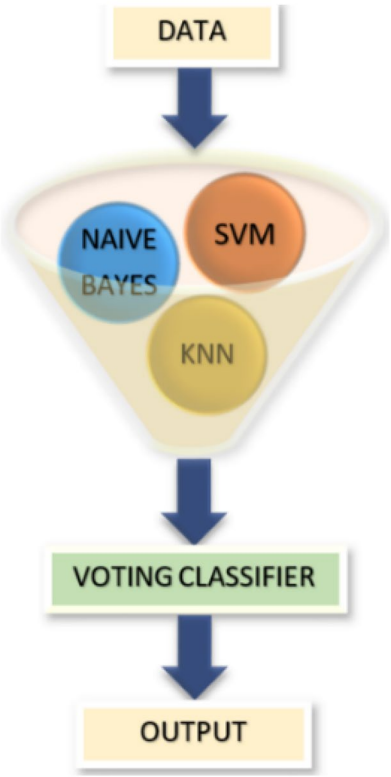


Fig. 1 Ensemble 1 using max voting technique

4.1.1 Voting classifier

In this study, three ensembles were made out of which one technique applied was max voting, which is a general ensemble technique [1]. Ensemble 1 comprises of a support vector machine (SVM), K—nearest neighbors (KNN), naïve Bayes at the inner layer. Voting classifier is used at the outer layer for dataset training. This study uses 10 K—fold validation as depicted in Fig. 1.

Algorithm 2: Ensemble 1 (Voting Classifier)

```
1. function VotingClassifiermodel
2.         xtrain,xtest,ytrain,ytest = SplitData(x,y,testsize=0.3)
3.         for each x∈F do
4.                   z = x()  [initialize]
5.                   models = append(z)
6.         endfor
7.         meta_learner = VotingClassifier(models)
8.         meta_learner = train(xtrain,ytrain)
9.         ypred = predict(xtest)
10.        accuracy = compare(ypred,ytest)
11.        return(ypred, accuracy)
12. endfunction
```

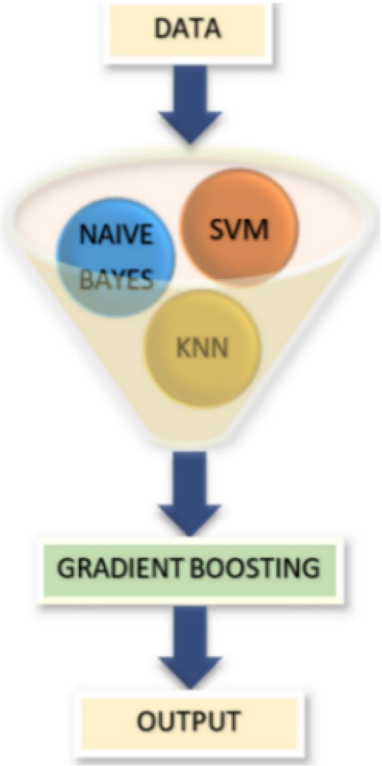


Fig. 2 Ensemble 2 using Gradient Boosting

4.1.2 Gradient boosting

Gradient boosting is a machine learning technique which is used for regression and classification problems [29]. It forms an ensemble by collecting all the weak models and obtain an accurate result.

Ensemble 2 was made using the technique of gradient boosting which is one of the advanced ensemble techniques. The ensemble consists of a support vector machine (SVM), K—nearest neighbors (KNN) at the inner layer. Including gradient classifier at the outer layer. The ensemble takes 1000 value as estimator with a learning rate of 0.005. Figure 2 shows the inner and outer layer of the ensemble 2 formed.

Algorithm 3: Ensemble 2 (Gradient Boosting)

```
1. function GradientBoostingmodel
2.         xtrain,xtest,ytrain,ytest = SplitData(x,y,testsize=0.3)
3.         xtrain_base,xtest_base,ytrain_base,ytest_base = SplitData(xtrain,ytrain,testsize=0.5)
4.         for each b∈F do
5.                   b = train(xtrain_base,ytrain_base)
6.                   z = predict(xtest_base)
7.                   ypred_base = append(z)
8.         endfor
9.         meta_learner= GradientBoostingClassifier(estimator=1000,maxdepth=3,lr=0.005)
10.        meta_learner = train(ypred_base,ytest_base)
11.        ypred = predict(xtest)
12.        accuracy = compare(ypred,ytest)
13.        return(ypred, accuracy)
14. endfunction
```

4.1.3 AdaBoost

AdaBoost is a meta—estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that the subsequent classifiers focus more on difficult cases [16]. This is considered to be one of the most accurate and precise algorithms among all the others.

Ensemble 3 was made using the advanced technique of ensemble called boosting which contains a technique named AdaBoost. This ensemble consists of a support vector machine (SVM), K—nearest neighbors (KNN), Naïve Bayes at the inner layer with AdaBoost at the outer layer. The model has been trained using a learning rate of 0.005 with 1000 as the estimator. As shown in the Fig. 3, the model uses SVM, KNN as the base and AdaBoost as the training classifier.

Algorithm 4: Ensemble 3 (AdaBoost Classifier)

```
function AdaBoostmodel
    xtrain,xtest,ytrain,ytest = SplitData(x,y,testsize=0.3)
    xtrain_base,xtest_base,ytrain_base,ytest_base = SplitData(xtrain,ytrain,testsize=0.5)
    for each b∈F do
        b = train(xtrain_base,ytrain_base)
        z = predict(xtest_base)
        ypred_base = append(z)
    endfor
    meta_learner = AdaBoostClassifier(estimators=1000,lr=0.005)
    meta_learner = train(ypred_base,ytest_base)
    ypred = predict(xtest)
    accuracy = compare(ypred,ytest)
    return(ypred, accuracy)
endfunction
```

4.1.4 XGBoost

XGBoost is an optimized version of gradient boosting which is highly efficient, flexible and portable [3]. It helps in implementing machine learning algorithms under the gradient boosting framework. It is relevantly fast and an accurate way to solve various problems (Table 2).

4.2 Supervised learning models

Supervised learning models are based on machine learning which uses the mapping of functions to make predictions by training the machine and then testing it on various inputs. There are various machine learning models which can be used in this study are as follows:

1) Support vector machines (SVM)
2) Naïve Bayes
3) K—nearest neighbors (KNN)

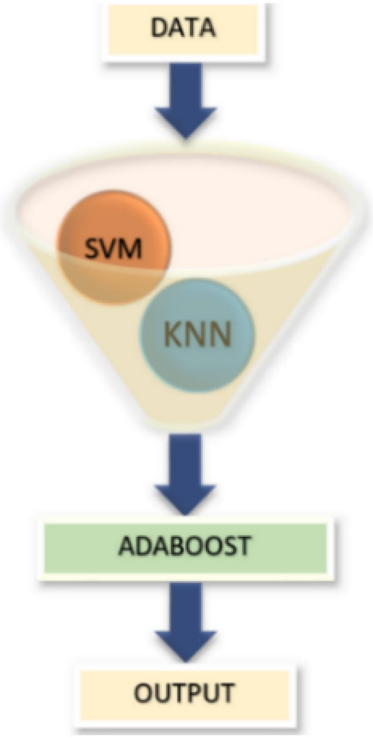


Fig. 3 Ensemble 3 using Ada-Boost Classifier

4) AdaBoost
5) Random forest
6) Ensemble learning algorithms
7) Neural network algorithms

In this study, these supervised learning algorithms were studied and implemented to train the machine and test on the dataset to predict the success rate.

4.2.1 Support vector machines (SVM)

Support vector machines algorithm is a supervised learning model which is associated with a learning algorithm that analyzes data used for classification and regression analysis [25]. This algorithm is used to solve both the classification and regression problems.

4.2.2 Naïve Bayes

Naïve Bayes algorithm is a classifier algorithm which is based on Bayes' theorem with an assumption of independence among predictors [26]. It assumes that the presence of features present in a class is not related to the presence of any other feature.

Table 2 List of attributes of the dataset obtained

Feature	Description	Data type
IMDb—ID	Uniquely identifies each movie	Text
Title	Movie's name	Text
IMDb—rating	Movie's rating on IMDb	Numerical
Release year	Year of release of the movie	Numerical
Release date	Date of Release of the movie	Date
Genre	Genre of the movie	Text
Writers	Writers of the movie	Text
Actors	Actors of the movie	Text
Directors	Directors of the movie	Text
Duration	Duration of the movie	Numerical
Production company	Production house of the movie	Text
Budget	Budget of the movie	Numerical
Net gross	Total earning of the movie	Numerical
Hit flop	Whether a movie is hit or flop	Categorical

4.2.3 K—nearest neighbors (KNN)

K—nearest neighbors is another algorithm which is used to solve both classification and regression problems. KNN algorithm assumes that similar things exist in close proximity [8].

4.2.4 Random forest

Random forest is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [15]. It is also a classifier for the test and training of the model. It is a classifier that built many classification trees as a forest of random decision trees, each constructed using a random subset of the features. In this study, 100 trees are used in the random forest algorithm.

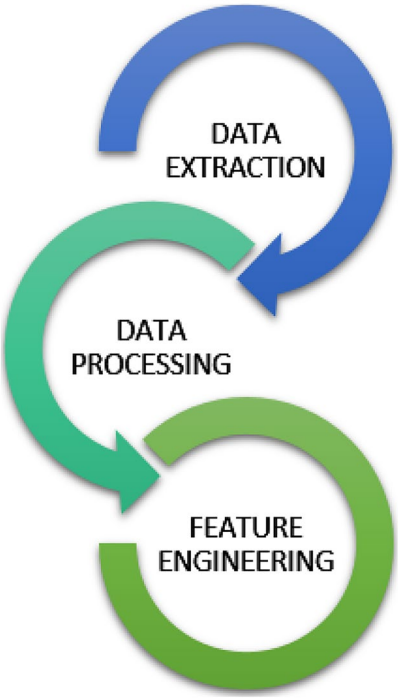
4.3 Neural network algorithms

Neural networks are one of the learning algorithms used within machine learning. It consists of different layers to predict and analyze the data [12]. Multilayer perceptron (MLP) is a class feedforward artificial neural network (ANN) [30]. It has various features and many hidden layers in it. In this study, four hidden layers with 30 features each were used to predict the movie success rate.

5 Methodology

In order to predict the success rate of a movie, a proficient dataset is required to get more accurate and precise results. The dataset used in this study was taken from Internet Movie Database (IMDb) and Box Office India (BOI) which contains all the attributes

Fig. 4 Dataset flowchart



like the title, actor, director, writer, genre, month of release, budget, IMDb rating, duration etc. These were the attributes which a usual dataset contains. So, in order to obtain results, the dataset was needed to be properly transformed and then implemented. Here, is an overview of how the dataset was made and then the methods used in this study.

5.1 Dataset description

The dataset initially contained 81,273 records in it. There were three stages to extract this data.

These three stages are:

1) Data extraction.
2) Data processing.
3) Feature engineering.

Figure 4 depicts the sequence followed for the formation of the dataset in which the dataset was first extracted, then processed to retrieve the useful information. On processing, the data was then transformed which contained only those attributes which are further used for the implementation of various models. Initially, the dataset contained many missing values and other challenges that are discussed in Sect. 4.3 which was a significant challenge for the movie prediction accuracy. These obstacles were resolved using various data transformation techniques which helped to prepare the final dataset.

5.1.1 Data extraction

At the initial stage, the dataset was extracted from IMDb[2] and BOI[3] websites [28]. The dataset contained different attributes like the title, IMDb—id, genre, duration, date of release, month of release, actor/actress, directors, writers, producers, co-actors, IMDb rating, BOI rating, budget, etc. At that point, the individual dataset from IMDb and BOI was consolidated to shape one dataset. The extracted dataset then comprised of 81,273 movies having 22 features, which were from Bollywood, Hollywood as well as South Film Industry.

5.1.2 Data processing

Once the dataset is analyzed, it is preprocessed in order to remove redundant dataset [4]. The dataset contained both Hollywood as well as Bollywood movies. As this study is completely based on Bollywood movie success rate prediction, all the Hollywood dataset was removed. The dataset then comprised of 5826 movies out of which movies released after 2000 were taken into consideration. After removing all the duplicate entries, the dataset now contained various non—common attributes like the BOI rating, co-actors, etc. These attributes were removed from the dataset in order to remove inconsistency. Thus, the merged dataset was processed so as to remove all the duplicate records as well as those columns which were not common were removed from the dataset. This gave an improved dataset which contained 1951 movies data.

5.1.3 Feature engineering

After processing the dataset, various transformations were made, i.e., some attributes like the date of release, producers, writers were removed with the goal that the study could concentrate on the key features that assume a significant job in predicting movie success [32]. After this the final dataset consisted of 1951 movies with 14 different features.

The next challenge was to fill up the missing values in the dataset in order to maximize the success rate. To overcome this challenge the dataset was first visualized for all the missing values by plotting graphs between features and budget of the movies. At that point the missing values were ordered into two classes: a portion of the missing values in the dataset was obliged by the mean calculated from the complete dataset. While, the staying missing values were filled taking mode according to the individual characteristics.

Further one hot encoding was done on the dataset and scaling was done after studying the variation of different attributes. The next step was to find out the key attributes that affect the outcome of the models. This was done with the help of correlational heat map. The heat map depicts the relationship among the different attributes that have the highest correlation among themselves. The final features that were considered are shown in the heat map along with their correlation. At the final stage of the dataset five attributes, genre, month of release, IMDb rating, duration and budget were considered. The final dataset thus obtained contained 1951 movies with 5 key attributes.

[2] https://www.IMDb.com/

[3] https://boxofficeindia.com/

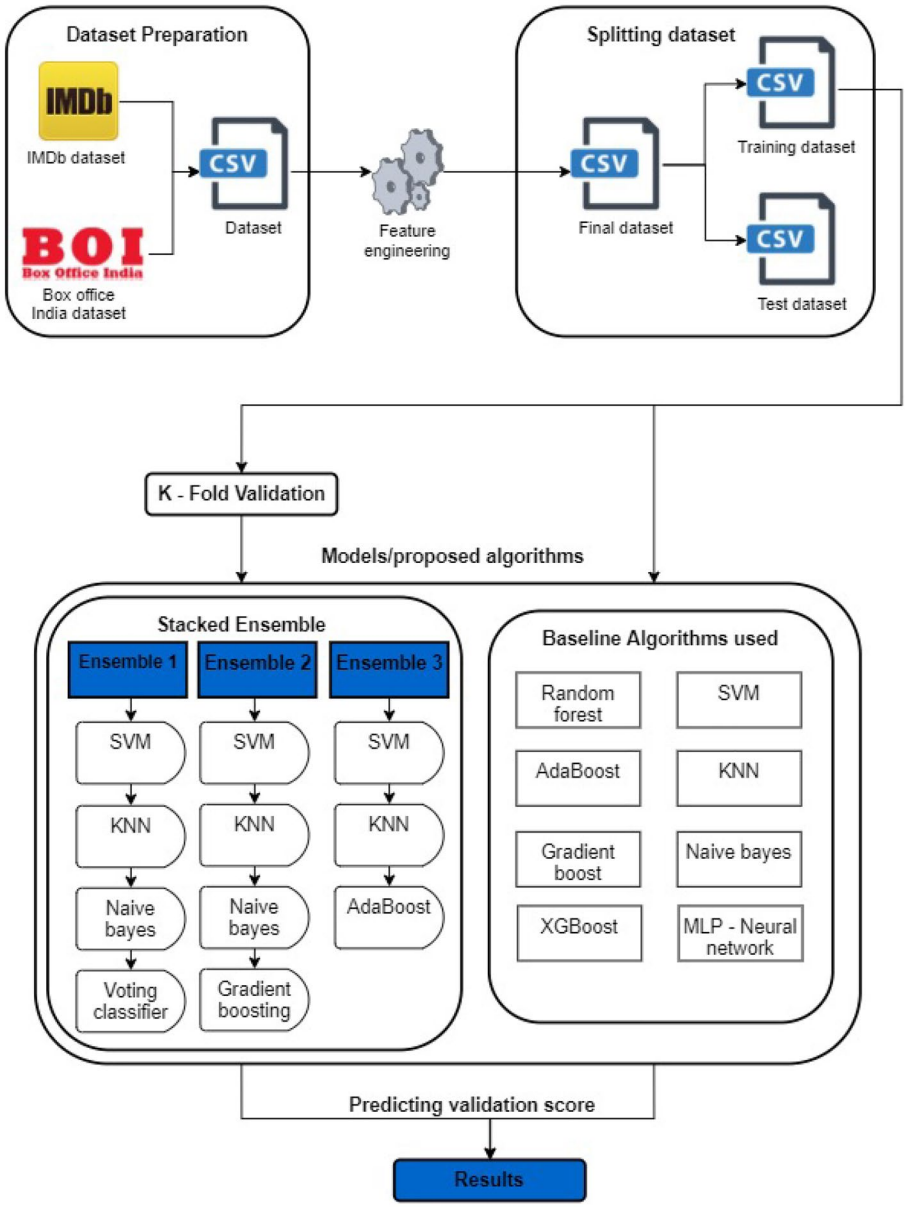


Fig. 5 Overview flowchart

After obtaining the final dataset, the next task was to implement the study of different algorithms on the dataset to obtain results for the movie prediction. Figure 5 shows the overall flowchart of the complete study and gives an idea of how various machine learning algorithms and ensemble models were implemented on the dataset in the sequential order. The general portrayal of different advances done in this study is shown in Fig. 5.

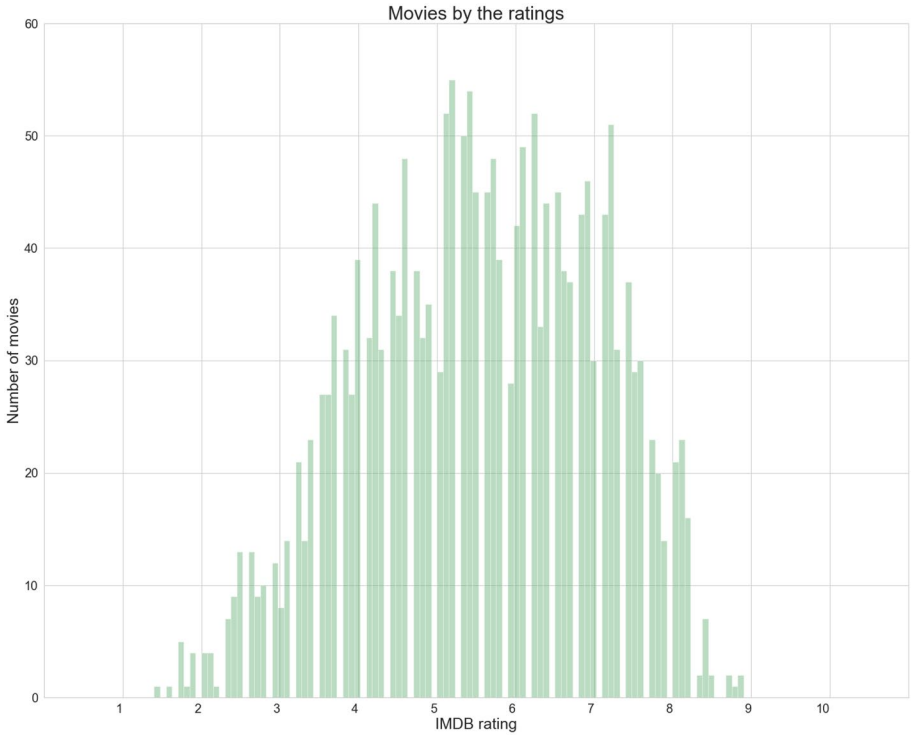


Fig. 6 Plot on the basis of rating

6 Results and discussion

Initially, the prepared dataset has been engineered to the best features as shown in Figs. 6, 7, 8, 9, 10. The analysis drawing various insights to understand the impact of taken factors on success of the movie is discussed below.

Figure 6 depicts the variation of movie's IMDb rating that were released between the year 2000 and year 2019. Rating is one of the most important attributes in predicting the movie success rate as people reviews are the most important to train our models and then implement them for the prediction. The variation in rating was used in scaling the attribute of the dataset which is therefore, one of the most important attributes in predicting the success of a movie.

Figure 7 shows the variation of number of movies released in different months. The Figure shows the highest movie releasing month is September while least is December. This concludes that month of release can be one such attribute which can be used to predict the movie success rate. As the greater number of movies released in a particular month as well as near the festivals will attract a lot of audience. The audience will therefor play an important factor in the movie success prediction.

Figure 8 shows the number of movies and the genres associated with them. Multiple genres were associated with a single movie. Genre is the key attribute to define any movie in a simplest manner. This attribute helps generate interest and excitement among

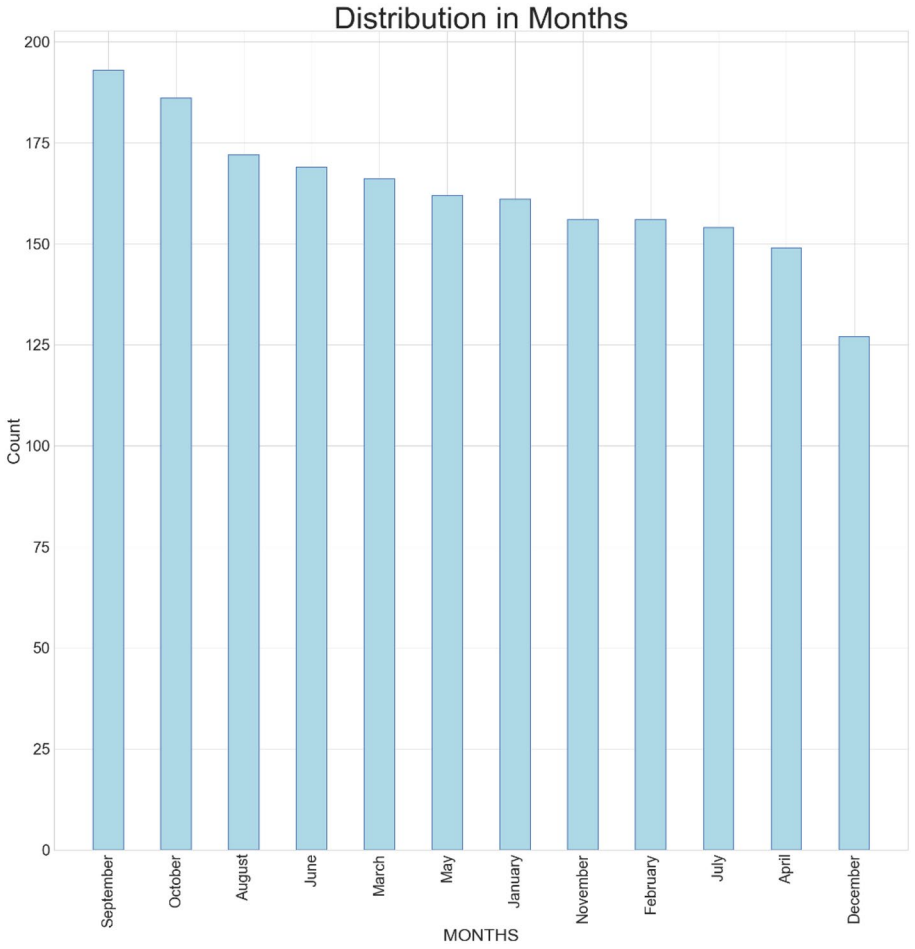


Fig. 7 Plot on the basis of Month

the audience for the movie. Hence, this attribute also served a measure role in the movie success prediction. Therefore, it has been taken into consideration.

Figure 9 represent the duration of movies with respect to number of movies from the dataset. Duration was indeed one of the major factors that affected the accuracy of the applied machine learning models and ensembles. Therefore, it has been considered as another major attribute for the success prediction.

After considering the all attributes such as month of release, duration, IMDb rating, genre, budget, actors, directors etc. correlational heat map as shown in Fig. 10 has been structured which showed the best output using the six major attributes which are month of release, IMDb rating, duration, genre, budget, hit/flop.

The above Fig. 10 which shows the correlation heat map depicts that when a correlation heat map of attributes like month, genre, IMDb rating, budget, hit/flop has been plotted, best results and combination was obtained out of the complete dataset.

With the help of these plots and correlational heat map, the task of scaling was done easily. A correlation matrix was then created which shows the relation among different

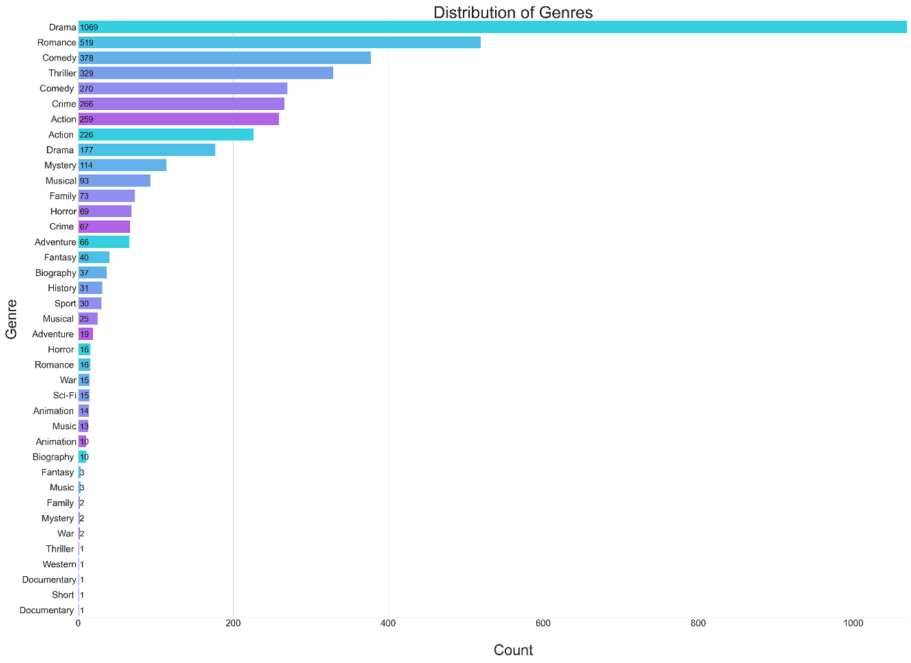


Fig. 8 Plot on the basis of Genre

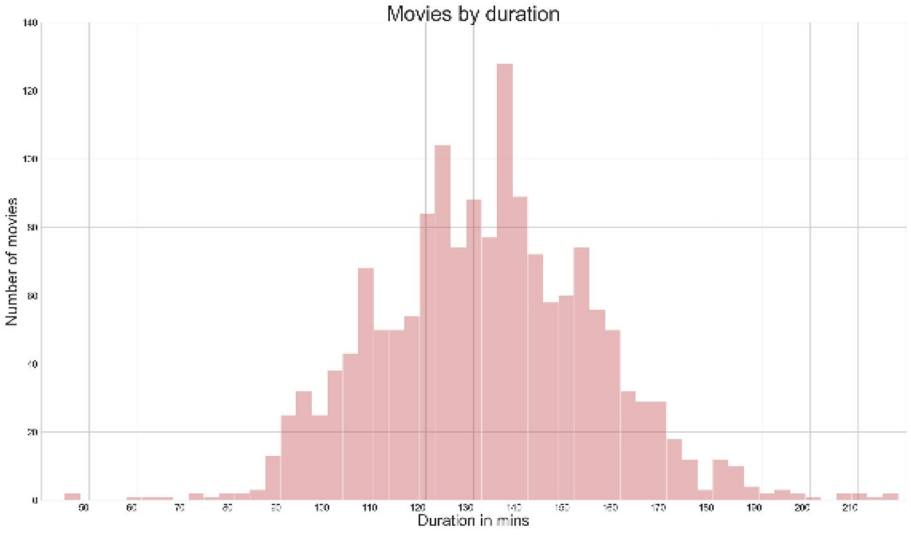


Fig. 9 Plot on the basis of Duration

machine learning algorithms as shown in Fig. 11. This correlation matrix was the base in order to choose the best suited algorithms for different ensembles techniques. The best combinations have been studied and then implemented (Fig. 12).

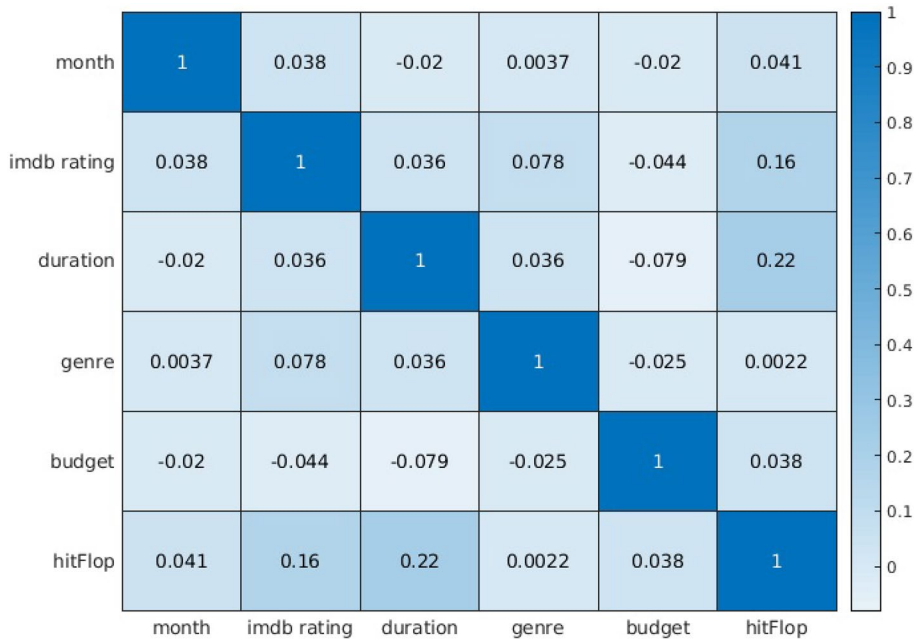


Fig. 10 Correlational heat map of the attributes

In this study, various algorithms were applied. The dataset was divided into train and test for the implementation of various algorithms. The proposed ensemble's training time varied between 3 and 5 s. The results provided in the section deals with the confusion matrix of the model along with the ROC curve obtained after implementing the model. The confusion matrix is a table that tells the statistical outcome of a model. The diagonal elements represent the true-positive value (TP) and the true-negative value (TN), while the other two represent the false-positive value (FP) and false-negative value (FN) [18]. Thus, the accuracy score can be calculated by the formula marked as Eq. (1).

The receiver operating characteristics or ROC curve [11] is a graphical analysis of the two values i.e., true-positive rate, which is also referred to as "sensitivity" which can be calculated from Eq. (2) and false-positive rate, which is referred to as "1-specificity" as shown in Eq. (3). The specificity and sensitivity have an inverse relationship i.e., one increases with a decrease in others and vice-versa. The area under the curve (AUC) is a characteristic of the ROC curve that depicts the precision of the model applied. The more curve shifts to the top left corner, the AUC value increases subjecting more true values resulting in increased accuracy of the model.

$$\text{Accuracy Score} = \frac{\text{Correct Predictions}}{\text{All Predictions}} = \frac{TP + TN}{TP + FN} \tag{1}$$

$$TPR = \frac{True\ Positive}{All\ Actual\ Positive} = \frac{TP}{TP + TN} \tag{2}$$

$$FPR = \frac{False\ Positive}{All\ Actual\ Negative} = \frac{FP}{TN + FP} \tag{3}$$

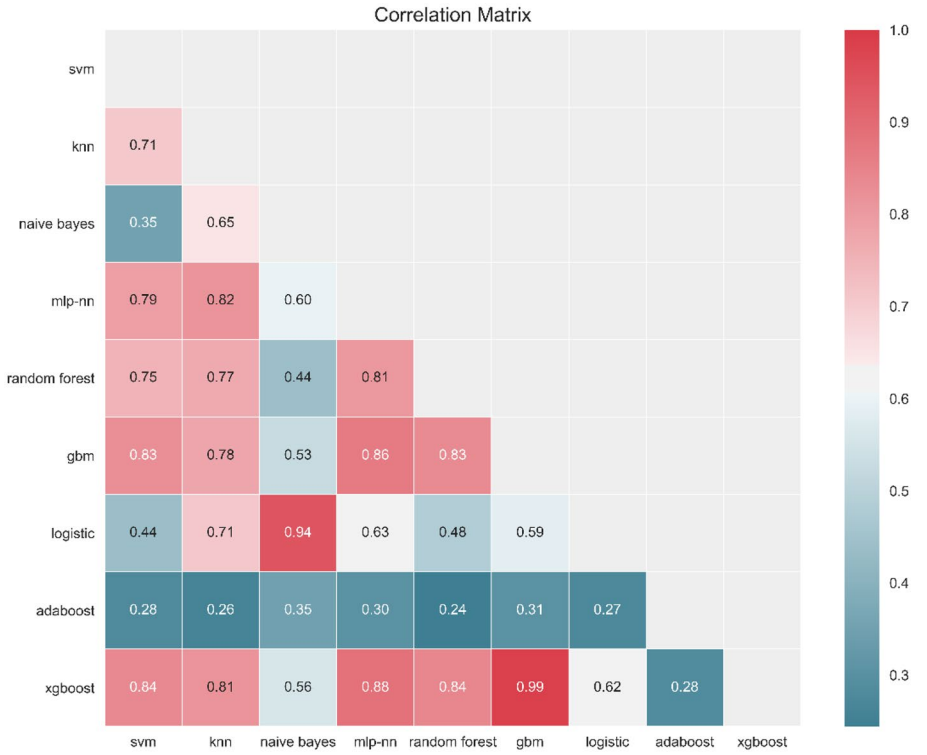


Fig. 11 Correlational matrix for different models

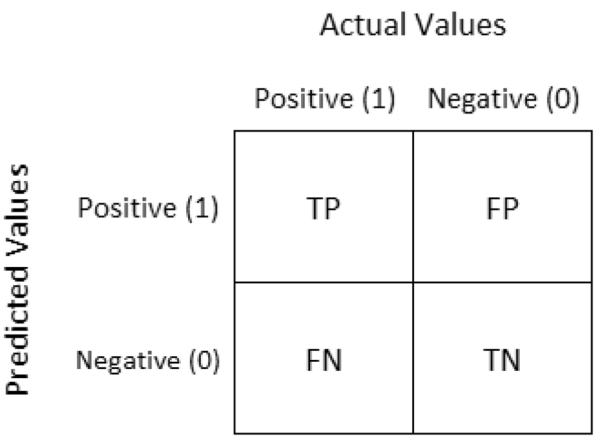


Fig. 12 Structure of confusion matrix

The error approximation in applied models have been calculated in three forms which are mean average error (MAE), mean square error (MSE) [19] and root mean square error (RMSE). MAE or mean average error depicts the average error of the predicted values and the original values taking magnitude of the difference found which can be found in Eq. (4).

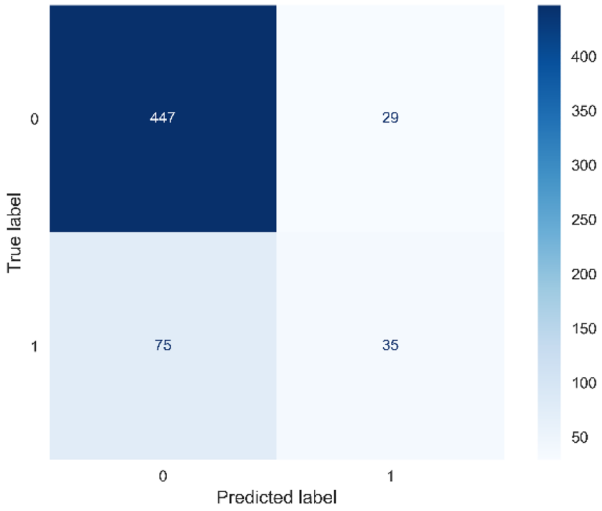


Fig. 13 Confusion matrix of SVM

$$MAE = \frac{1}{n} \sum | y - \hat{y}| \tag{4}$$

The MSE or mean square error is the error calculated by taking the mean of the square of difference of predicted and actual value and can be calculated from Eq. (5).

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2 \tag{5}$$

The RMSE or root mean square error [20] is preferred over MAE and MSE in case of large-scale errors. As the name suggests, RMSE is calculated by taking the square root of MSE as shown in Eq. (6). The RMSE gives a higher weight to large errors as it takes the square of predicted and actual values. The values which are closer to 0 are better as they all are negatively directed errors.

$$RMSE = \sqrt{MSE} \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n} \sum (y - \hat{y})^2} \tag{7}$$

K-Fold is a popular technique used to make comparatively a less biased model. It ensures that every subset of the dataset gets a chance to perform in both the training and testing section. K-Fold plays an important factor in terms of accuracy because it splits the dataset into k-sections on which training and testing are applied to each section. This results in equal contributions from every section of the dataset, which helps to get better accuracy.

Figures 13 and 14 represents the confusion matrix and ROC curve of the SVM model applied. The diagonal elements represent the true positive and true negative values while other elements show the false values in the prediction. The ROC curve prepared shows that the prediction output is just satisfactory.

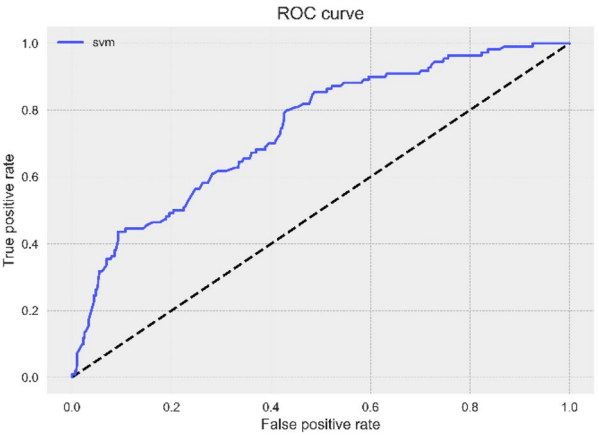


Fig. 14 ROC curve of SVM

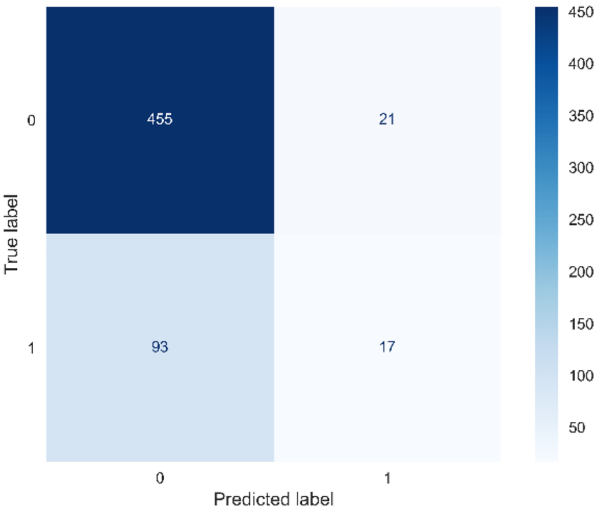


Fig. 15 Confusion matrix of KNN

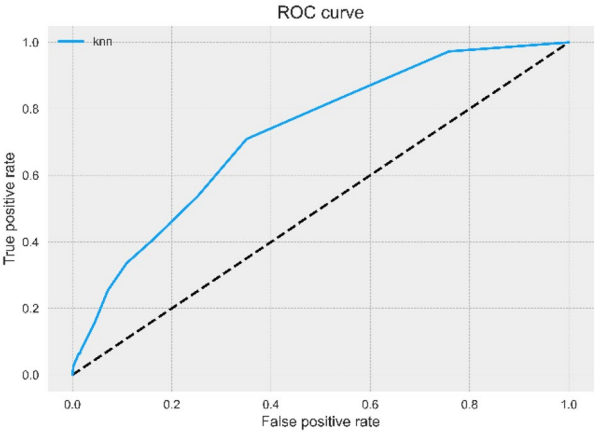


Fig. 16 ROC curve of KNN

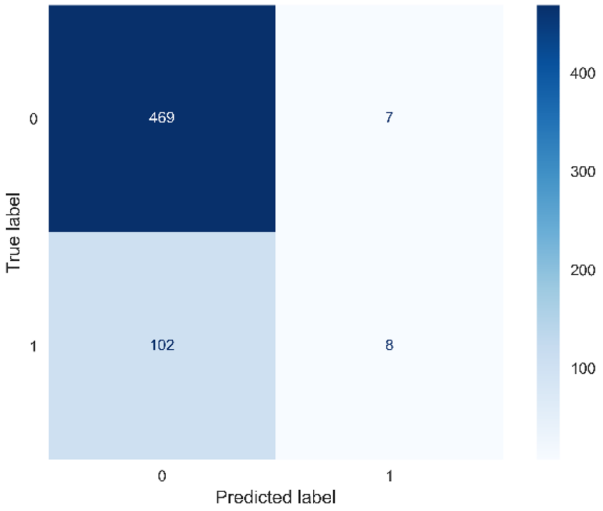


Fig. 17 Confusion matrix of Naïve Bayes

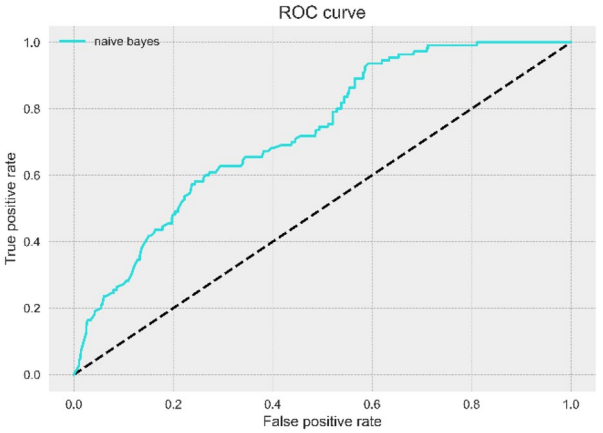


Fig. 18 ROC curve of Naïve Bayes

Figures 15 and 16 represent the confusion matrix and ROC curve of KNN model applied in the study. The confusion matrix was created on the prediction by KNN model on a subset of the dataset. The ROC curve depicts that the model applied is not satisfactory. The more curve moves to the left top corner, the more accurate predictions are.

Figures 17 and 18 represent the confusion matrix and ROC curve of Naïve Bayes machine learning model applied in the study. The truth positive predicted values obtained in this model are quite better than the others and ROC curve with moderate sensitivity and specify.

Figures 19 and 20 shows the confusion matrix and ROC curve of the random forest model. The true positive and true negative values were significantly good for making of the ROC curve which shows better outputs among other supervised learning algorithms.

The results of the baseline models like KNN, SVM and various other models, which have been applied in the study gave the précised results but further ensembles were applied

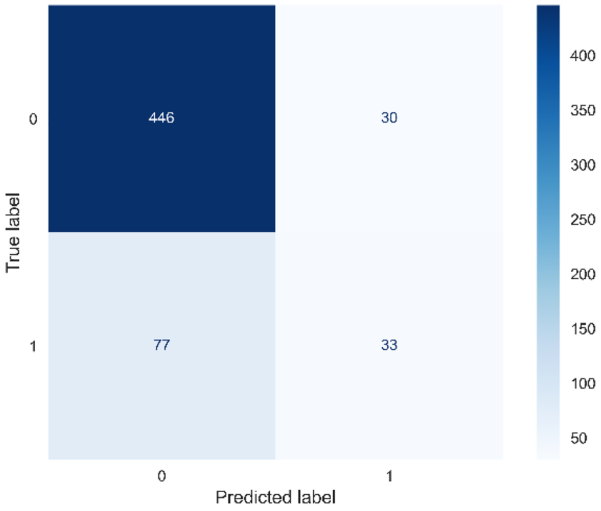


Fig. 19 Confusion matrix of random forest

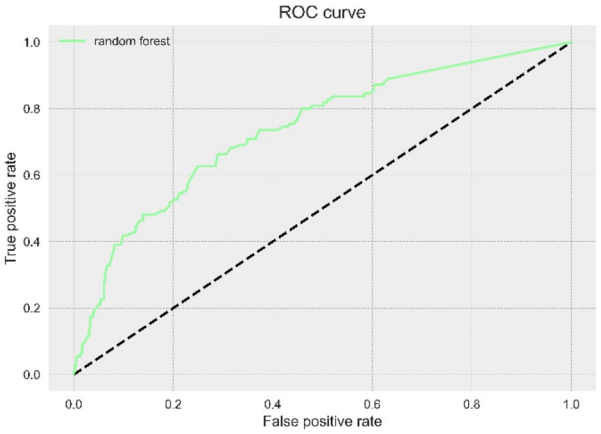


Fig. 20 ROC of random forest

to increase the accuracy. The accuracy of the ensembles came out to be far better than the baseline models applied. An increase in AUC value along with improved ROC curve was also observed in the further results as shown below.

Figures 21 and 22 represent the confusion matrix and ROC curve of gradient boosting model. The confusion matrix shows the best results with highest percentage of true values in comparison with other models. The ROC curve shown above depicts the high accuracy of the model with high sensitivity and specificity.

Figures 23 and 24 depict the confusion matrix and ROC curve of AdaBoost model of machine learning. The figure illustrates that the results are much better than other models with a high AUC value as interpreted by ROC curve which leads to the improved accuracy.

The increased true-positive values in the confusion matrix results in increased sensitivity in the ROC curve, and a higher cut-off value. The cut-value in ROC curve is the point where "sensitivity + specificity-1" is maximum.

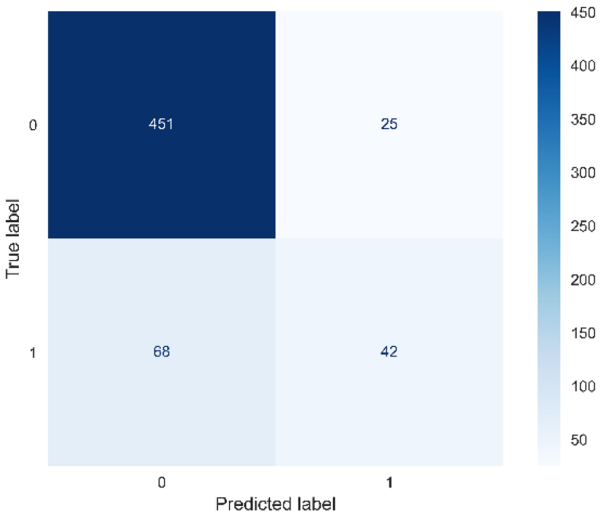


Fig. 21 Confusion matrix of GBM

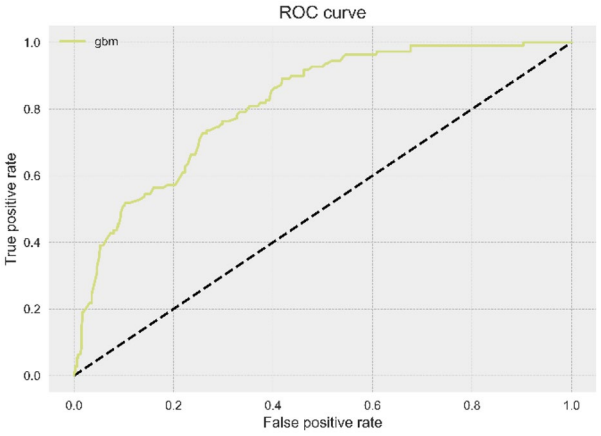


Fig. 22 ROC curve of GBM

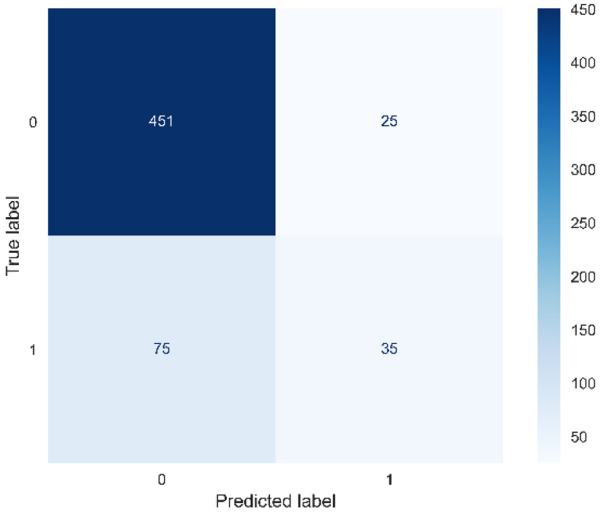


Fig. 23 Confusion matrix of AdaBoost

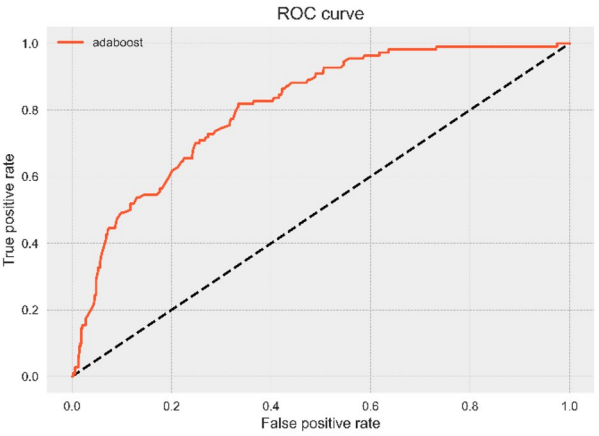


Fig.24 ROC curve of AdaBoost

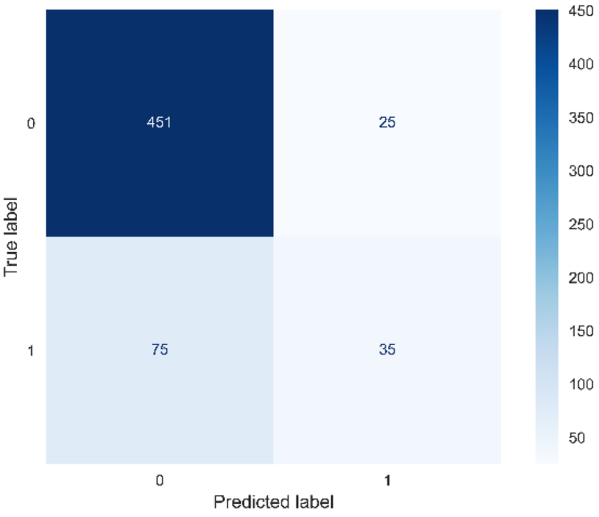


Fig. 25 Confusion matrix of XGBOOST

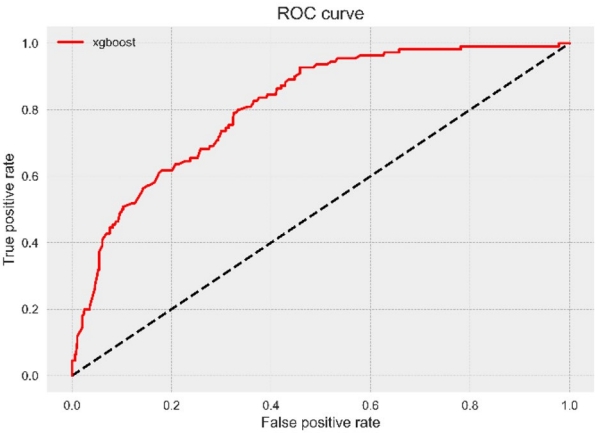


Fig. 26 ROC curve of XGBOOST

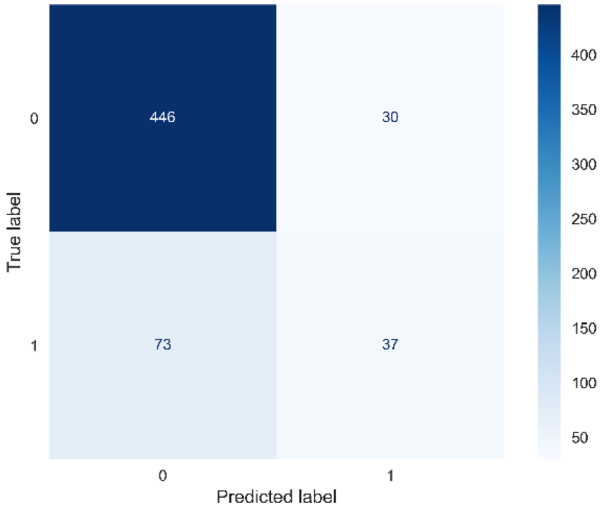


Fig. 27 Confusion matrix of MLP—NN

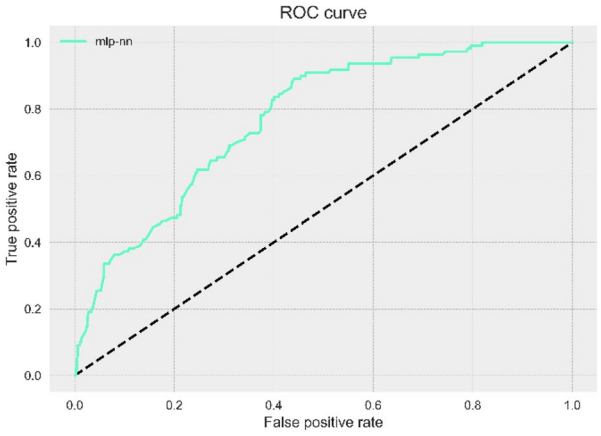


Fig. 28 ROC curve of MLP—NN

Figures 25 and 26 shows the confusion matrix and ROC curve of XGBoost model. The confusion matrix shown above and confusion matrix of AdaBoost came out to be similar, therefore showing better predictions than the other machine learning models. The ROC curve prepared was also similar to that of AdaBoost model with high cut-off value.

Figures 27 and 28 shows the confusion matrix and ROC curve of multilayer perceptron neural network (MLP-NN) model. The valued obtained in confusion matrix were satisfactory which gives an average ROC curve with moderate sensitivity which in turn result in a little increase in specificity.

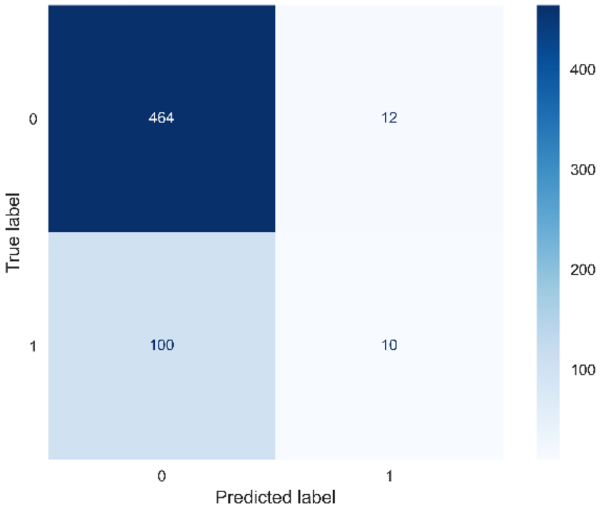


Fig. 29 Confusion matrix of Ensemble 1

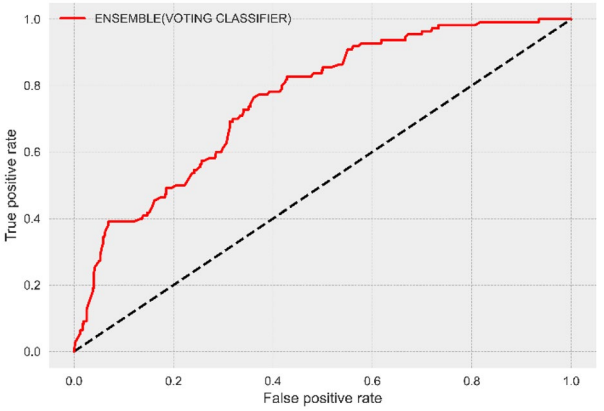


Fig. 30 ROC curve of Ensemble 1

Figures 29 and 30 represent the confusion matrix and ROC curve of the ensemble 1 using voting classifier with 3 machine learning models i.e., SVM, Naïve Bayes, and KNN. The ROC curve shown depicts the moderate accuracy of the model applied.

Figures 31 and 32 shows the confusion matrix and ROC curve of ensemble 2 combining SVM, Naïve Bayes and KNN using gradient boosting classifier. The ROC curve shown has less area under the curve, showing average results.

Figures 33 and 34 shows the confusion matrix and ROC curve of ensemble 3 combining SVM and KNN models using AdaBoost classifier. The results obtained by the confusion matrix and ROC curve are middling with moderate accuracy.

Thus, after making the confusion matrix and the ROC curves of all the algorithms were studied and combining all ROC a single plot was made as shown in Fig. 35.

On implementing these algorithms on our dataset and studying the ROC curves. Table 3 shows the accuracy and comparison of all algorithms.

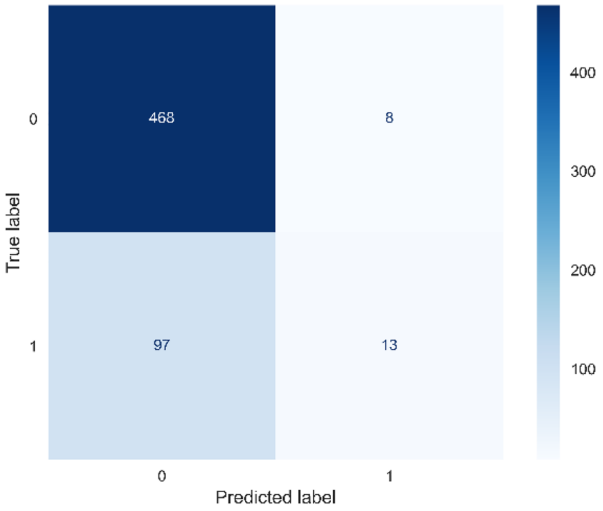


Fig. 31 Confusion matrix of Ensemble 2

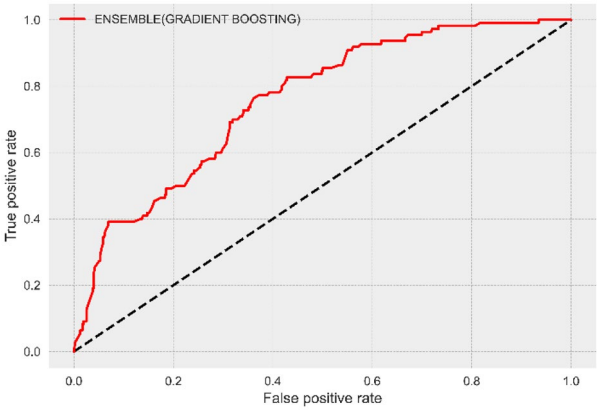


Fig. 32 Roc curve of Ensemble 2

In recent times, various studies have been proposed on movie prediction. The different studies predicted the success of the movies by taking different set of attributes best suitable for the applied models. The dataset varied in different works, but our study approached with better outcomes working with the dataset of 1951 movies. A comparative analysis with previous works on the same subject is shown in the Table 4.

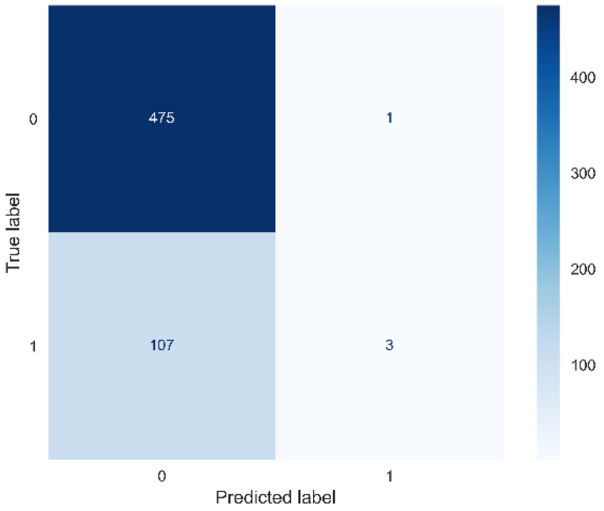


Fig. 33 Confusion matrix of Ensemble 3

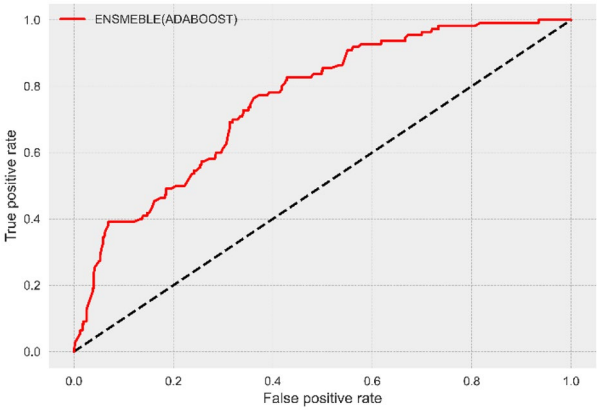


Fig. 34 ROC curve of Ensemble 3

In this study, gradient boosting gave us the maximum results. Results were also obtained by using various ensemble learning techniques. XGBoost was also used to train and test the dataset which gave an accuracy of 83.54% after applying 10-fold validation. Table 3 also shows the accuracy and AUC of different algorithms with and without folds for better results, both the accuracies were calculated and compared.

The maximum accuracy is obtained by using gradient boosting i.e., 84.1297% without folds and 83.6518% after applying tenfold validation as shown in Table 3.

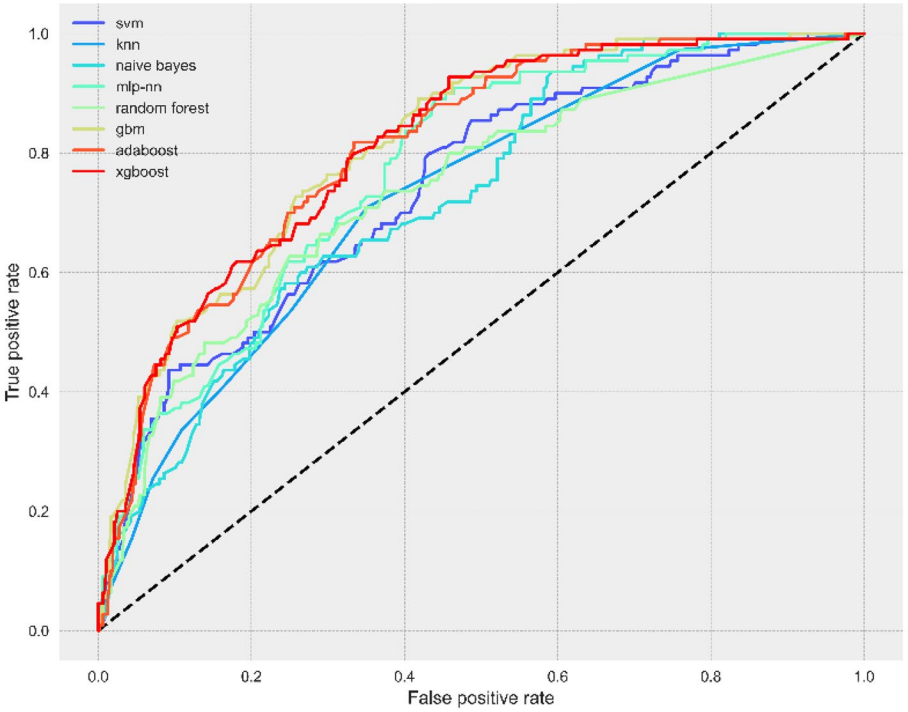


Fig. 35 Combined ROC curve of all the algorithms

Table 3 Results obtained from various models/algorithms

Model	Accuracy (%)	AUC score	RMSE	MSE	MAE	10-Fold cross validation accuracy	10-Fold cross validation AUC score
SVM	81.9112	0.741	0.4253	0.1809	0.1809	82.7804	0.7475
KNN	80.5461	0.725	0.4411	0.1945	0.1945	81.1915	0.7415
Naïve- Bayes	81.3993	0.728	0.4313	0.1860	0.1860	81.5502	0.6900
MLP—neural network	81.2287	0.723	0.4332	0.1877	0.1877	80.7813	0.6952
Random forest	81.7406	0.737	0.4273	0.1826	0.1826	80.5759	0.7564
Gradient Boosting	84.1297	0.815	0.3984	0.1587	0.1587	83.6518	0.7991
AdaBoost	82.9352	0.808	0.4131	0.1706	0.1706	82.9853	0.7948
XGBoost	82.9352	0.812	0.4131	0.1706	0.1706	83.5492	0.8044
Ensemble 1	81.5700	0.775	0.4293	0.1843	0.1843	81.3451	0.7700
Ensemble 2	81.9113	0.769	0.4253	0.1809	0.1809	83.4461	0.8060
Ensemble 3	81.5700	0.783	0.4293	0.1843	0.1843	80.8838	0.7561

Table 4 Table of comparison with existing work

	Proposed Approach	Khandelwal et al. [32]	Masih et al. [18]	Kanitkar [11]	Quader et al. [19]	Nithin et al. [20]
MAE	0.1587	–	–	22.2	–	–
MSE	0.1587	–	–	1541.19	–	–
RMSE	0.3984	–	–	39.2	–	–
AUC Score	0.815	0.809	0.66	0.62	0.51	–
Accuracy (%)	84.1297	–	–	49.33	58.53	51

7 Conclusion

The cinema industry is one of the world's fastest-growing industries. Predicting a film's success at the box office before it's released is one of the crucial steps in the future film industry that will aid in growth and development. A film's popularity is not only dependent on film-related factors but also on audience opinions. In this research, we use different machine learning algorithms and ensembles to construct various models to predict a film's success rate. Initially, the data was extracted and processed, creating a final data set of 1951 movies with various attributes for prediction. KNN is the least successful algorithm for prediction with the success rate of 80.5461%. Gradient Boosting, AdaBoost and XGBoost shows great results with Gradient Boosting being the best of all having a success rate of 84.1297% and AUC 0.815. AdaBoost and XGBoost have the success rate of 82.9352% each and AUC 0.808 and 0.812 respectively.

For future improvements, more features like user reviews, actors, directors, etc., that would also have a great impact in predicting the success of a movie can also be considered. The prepared dataset can also increase, so better results would be obtained from the model. New ensembles will also be implementing in the future work to check the improvement of success rate. Check the improvement of success rate.

References

1. Bauer E, Kohavi R (1999) An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Mach Learn 36(1–2):105–139
2. Bhave A, Kulkarni H, Biramane V, Kosamkar P (2015) Role of different factors in predicting movie success. In: 2015 International Conference on Pervasive Computing (ICPC). IEEE, pp 1–4
3. Chen T, He T, Benesty M, Khotilovich V, Tang Y (2015) Xgboost: extreme gradient boosting. R package version 0.4–2, 1–4
4. Das AK, Kalam S, Kumar C, Sinha D (2021) TLCoV-An automated Covid-19 screening model using transfer learning from chest X-ray images. Chaos Solitons Fractals 144:110713
5. Dhir R, Raj A (2018) Movie success prediction using machine learning algorithms and their comparison. In: 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), IEEE, pp 385–390
6. Dietterich TG (2002) Ensemble learning. The handbook of brain theory and neural networks 2:110-125
7. Doshi L, Krauss J, Nann S, Gloor P (2010) Predicting movie prices through dynamic social network analysis. Procedia Soc Behav Sci 2(4):6423–6433
8. García-Laencina PJ, Sancho-Gómez JL, Figueiras-Vidal AR, Verleysen M (2009) K nearest neighbours with mutual information for simultaneous classification and missing data imputation. Neurocomputing 72(7–9):1483–1493

9. Jain V (2013) Prediction of movie success using sentiment analysis of tweets. Int J Soft Comput Softw Eng 3(3):308–313
10. Jernbäcker C, Pojan S (2017) Predicting movie success using machine learning techniques
11. Kanitkar A (2018). Bollywood movie success prediction using machine learning algorithms. In: 2018 3rd International conference on circuits, control, communication and computing (I4C). IEEE, pp 1–4
12. Kumar V, Das AK, Sinha D (2019) UIDS: a unified intrusion detection system for IoT environment. Evol Intell 1–13
13. Lash MT, Zhao K (2016) Early predictions of movie success: the who, what, and when of profitability. J Manag Inf Syst 33(3):874–903
14. Latif MH, Afzal H (2016) Prediction of movies popularity using machine learning techniques. Int J Comput Sci Netw Sec 16(8):127
15. Liaw A, Wiener M (2002) Classification and regression by randomForest. R News 2(3):18–22
16. Liu H, Tian HQ, Li YF, Zhang L (2015) Comparison of four Adaboost algorithm based artificial neural networks in wind speed predictions. Energy Convers Manage 92:67–81
17. Modi A, George EL (2020) Genre-based indian viewer movie reviews—A sentiment analysis classification of text and emoticons with a supervised machine learning approach. In: Advanced computing technologies and applications. Springer, Singapore, pp 633–644
18. Nikita J, Gupta V, Shubham S, Madan A, Chaudhary A, Santosh KC (2021) Understanding cartoon emotion using integrated deep neural network on large dataset. Neural Comput Appl 1–21
19. Nikita J, Jhunthra S, Garg H, Gupta V, Mohan S, Ahmadian A, Salahshour S, Ferrara M (2021) Prediction modelling of COVID using machine learning methods from B-cell dataset. Results Phys 21:103813
20. Nithin VR, Pranav M, Sarath B, Lijiya A (2014) Predicting movie success based on IMDb data. Int J Data Min Tech Appl 3:365–368
21. Piryani R, Gupta V, Singh VK (2017) Movie prism: a novel system for aspect level sentiment profiling of movies. J Intell Fuzzy Syst 32(5):3297–3311
22. Piryani R, Gupta V. Singh VK, Ghose U (2017) A linguistic rule-based approach for aspect-level sentiment analysis of movie reviews. In: Advances in computer and computational sciences. Springer, Singapore, pp 201–209
23. Pradeep K, TintuRosmin CR, Durom SS, Anisha GS (2020). Decision tree algorithms for accurate prediction of movie rating. In: 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC). IEEE, pp 853–858
24. Quader N, Gani MO, Chaki D, Ali MH (2017) A machine learning approach to predict movie box-office success. In: 2017 20th International Conference of Computer and Information Technology (ICCIT). IEEE, pp 1–7
25. Schlkopf B, Smola AJ, Bach F (2018) Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press
26. Tzanos G, Kachris C, Soudris D (2019) Hardware acceleration on gaussian naive bayes machine learning algorithm. In: 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST). IEEE, pp 1–5
27. Verma H, Verma G (2020) Prediction model for bollywood movie success: a comparative analysis of performance of supervised machine learning algorithms. Rev Socionetw Stratg, 14(1): 1–17. (biggest limitation that the dataset was used was very small of about 200 movies)
28. Watershed (Iran) using an artificial neural network model: a comparison between multilayer perceptron (MLP) and radial basic function (RBF) algorithms. Arab J Geosci 6(8):2873–2888
29. Xu Q, Xiong Y, Dai H, Kumari KM, Xu Q, Ou HY, Wei DQ (2017) PDC-SGB: Prediction of effective drug combinations using a stochastic gradient boosting algorithm. J Theor Biol 417:1–7
30. Zare M, Pourghasemi HR, Vafakhah M, Pradhan B (2013) Landslide susceptibility mapping at Vaz
31. Zhang W, Skiena S (2009) Improving movie gross prediction through news analysis. In: 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, vol. 1, IEEE, pp 301–304
32. Zheng A, Casari A (2018) Feature engineering for machine learning: principles and techniques for data scientists. O'Reilly Media, Inc

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.