Movie recommendation system using machine learning.

Authors:

Seif Eldin Hussein Mostafa Ali **E-mail address: seif.hussin11@gmail.com**

Khaled Waheed Mohamed Ali **E-mail address: waheed33603@gmail.com**

github https://github.com/Seifhussein1/Movie-Recommendation-System.git
OCTOBER SIX UNIVERSITY
FACULTY OF INFORMATION SYSTEMS AND COMPUTER SCIENCE (AI DEPARTMENT)

ARTICLE INFO

Keywords:
Cosine similarity
Movie recommendation
Naïve Bayes
Sentiment analysis
Support vector machine
KNN
Bagging
Decision Tree
Gradient Boosting
Perceptron
logistic Regression
Extra Trees Classifier
Random Forest
Ada Boost Classifier

ABSTRACT

In the modern world, where technology is at the forefront of every industry, there has been an overload of information and data. Thus, a recommendation system comes in handy to deal with this large volume of data and filter out the useful information which is fast and relevant to the user’s choice. This paper describes an approach to a movie recommendation system using Cosine Similarity to recommend similar movies based on the one chosen by the user. Although the existing recommendation systems get the job done, it does not justify if the movie is worth spending time on. To enhance the user experience, this system performs sentiment analysis on the reviews of the movie chosen using machine learning. Two of the supervised machine learning algorithms, Naïve Bayes (NB) Classifier and Support Vector Machine (SVM) Classifier, are used to increase the accuracy and efficiency. This paper also gives a comparison between NB and SVM on the basis of parameters like Accuracy, Precision, Recall, and F1 Score. The accuracy score of SVM came out to be 98.63% whereas the accuracy score of NB is 97.33%. Thus, SVM outweighs NB and proves to be a better fit for Sentiment Analysis. Additionally, we employed a diverse set of 10 classifiers to develop a robust movie recommendation system. Our analysis revealed that the K-Nearest Neighbors (KNN) classifier achieved the lowest accuracy of 96.749%, while the AdaBoostClassifier demonstrated the highest accuracy of 99.061%. Furthermore, we explored the combined performance of classifiers by pairing them and evaluating their effectiveness using a voting ensemble approach. Among the combinations, the KNN and Support Vector Classifier (SVC) pair exhibited the minimum accuracy of 97.471%. Conversely, the AdaBoostClassifier and ExtraTreesClassifier pair demonstrated the highest accuracy of 99.205%. Additionally, utilizing three classifiers, 'Decision Tree', 'ExtraTreesClassifier', and 'AdaBoostClassifier', showcases remarkable performance, with AdaBoostClassifier achieving a maximum accuracy of 99.422%. These findings underscore the significance of ensemble methods in enhancing the recommendation system's performance.

1. Introduction

Since its invention, the Internet has grown rapidly and continues to grow each day. The abundance of information available online, has made it a strenuous task to access the right information quickly and easily [1]. Fortunately, this problem can be solved with the help of recommendation systems.

Recommendation systems are used extensively today and have found applications in multiple industries such as e-commerce, retail, banking, entertainment etc. These systems collect and auto-analyse the user data to generate personalised recommendations for the users [2]. The most common approaches to implement recommendation systems are Content-based Filtering (CBF), Collaborative Filtering (CF) and Hybrid Filtering [3]. CBF is an approach that is used to analyse the content of each item and recommend other items that have similar characteristics. CF addresses some of the limitations of CBF and provides recommendations by comparing the similarities between the users and the items. It uses the knowledge of the user’s previous preferences as well as the preferences of other similar users to generate a recommendation. Many recommendation systems are also known to use the Hybrid-filtering technique combining the features of both CBF and CF methods [4].

A movie’s popularity is based on the type of reviews it gets from the audience . These reviews are also responsible for affecting the choice of other users. Users are more likely to choose a movie that was preferred by most people rather than a movie that was largely disliked [5]. Analyzing these reviews, ignoring the reviews that contain misleading information also adds to the difficulty of decision-making [6]. Sentiment Analysis provides a solution to this problem.

We apply 10 classifiers ' KNN’, ’SVC’, ‘Bagging’, ‘Decision Tree’,‘GradientBoostingClassifier’, ‘Perceptron’ , ‘logistic Regression’ ,‘ExtraTreesClassifier’ , ‘Random Forest’, ‘AdaBoostClassifier’

Sentiment Analysis facilitates a way to use NLP (natural language processing) to extract information from a textual source and classify the statement or word or document as positive or negative. It is very useful to understand the opinion of the author and indicate the user experience.

Opinion mining uses the concepts of data mining to extract and classify the opinions expressed in various online forums or platforms. This enables better understanding of the user’s sentiment or feeling towards a particular subject matter [7].

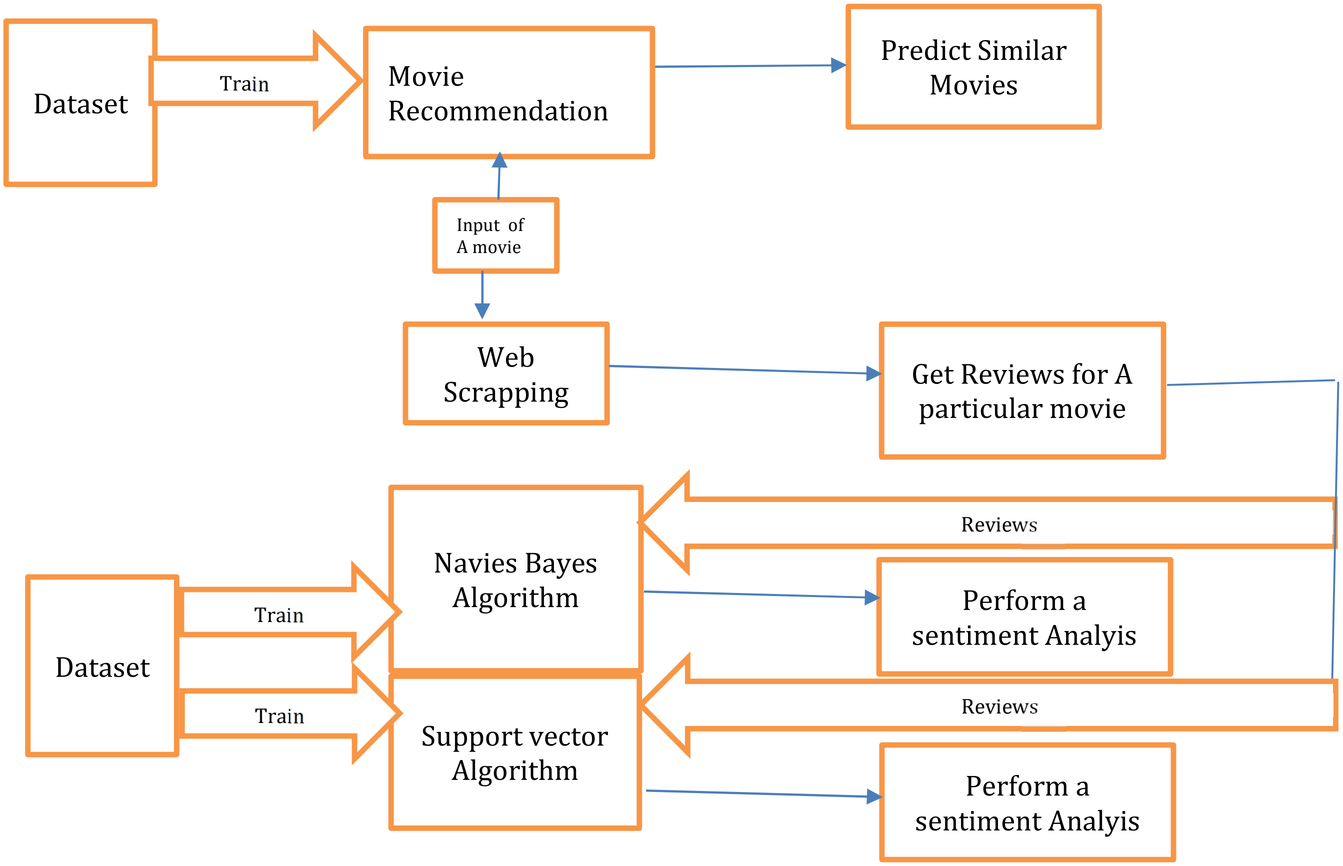
The paper presents a system that not only recommends movies to the users but also analyses and classifies the reviews into positive or negative. The movie recommendation part is performed using Cosine Similarity and a comparison is drawn between SVM and the NB algorithm to perform the Sentiment Analysis of the reviews.

The objective of the study is to deal with the large volume of data and filter useful information, recommend similar movies based on user's choice and perform Sentimental Analysis on the reviews of the movie chosen.

we are modifying in preprocessing in converting textual data into numerical representations (TF-IDF vectors) and then calculate the similarity between documents based on their TF-IDF representations using cosine similarity. In addition to utilizing 10 classifiers, this paper evaluates their performance individually. The results indicate that the K-Nearest Neighbors (KNN) classifier achieved the lowest accuracy of 96.749%, while the AdaBoostClassifier demonstrated the highest accuracy of 99.061%. Furthermore, combining different classifiers in pairs using a voting ensemble approach revealed interesting insights. For instance, the combination of KNN and SVC yielded the minimum accuracy of 97.471%, whereas the combination of AdaBoostClassifier and GradientBoostingClassifier achieved the highest accuracy of 99.205%. Additionally, utilizing three classifiers, 'Decision Tree', 'ExtraTreesClassifier'and'AdaBoostClassifier',showcases remarkable performance, with AdaBoostClassifier achieving a maximum accuracy of 99.422%.so the results of what we do is better than the old one, then we recommend to use our methodologies.

The paper follows the given structure; Section 2 covers the Literature Review. Section 3, discusses the Methodology which includes the dataset, the pre-processing of data, mining of data for movie recommendation, machine learning for sentiment analysis and finally, the performance report. Section 4 comprises the results and discussions. Section 5 contains pseudocode. Representing workflow in Section 6 using flowchart. Finally, the conclusion can be found in Section 7 And References in Section 8.

Fig. 1. Flowchart of proposed method.



2. Related works

In this section, the various existing methods and the drawbacks of the existing work are discussed in detail.

In this paper, the authors propose a hybrid approach that combines a content-based approach with genre correlation to implement a recommendation system. This system takes into account both the user ratings of the movie as well as their genres while making recommendations to the user [4].

In this paper, a new similarity algorithm is introduced. This is called User Profile Correlation-based Similarity (UPCSim) and it allows other user behavioral data to influence the accuracy of the recommendation.It calculates the weights of similarity which depend on the user's rating and the user's behaviour value and classifies the preferences of the user using K-nearest Neighbours algorithm. While this approach shows a de- crease in the Mean Absolute Error (1.64%) and the Root Mean Square Error (1.4%), it requires more computation time [8].

The authors discuss the implementation of a movie recommendation system using two algorithms, Cosine Similarity and K-Nearest Neighbours. The movie recommendation is done using the cosine similarity algorithm. A normalized popular score is used to obtain the function for computing distance and the K-Nearest Neighbours algorithm is applied to enhance the accuracy [9].

In this paper, a hybrid recommendation system is proposed that uses sentiment analysis of user tweets for movies to obtain a sentiment score to improve the recommendation made to the users using a weighted fusion score method [10].

This paper implements five machine learning classifiers – Multinomial Naïve Bayes, SVM, Decision Tree, Bernoulli Naïve Bayes, Maximum Entropy are applied on the pre-processed data containing feature vectors to classify the movie reviews data [11].

Considering the utilization of 10 classifiers, this paper assesses their individual accuracies. Notably, the K-Nearest Neighbors (KNN) classifier achieves the lowest accuracy of 96.749%, while the AdaBoostClassifier demonstrates the highest accuracy of 99.061%. Furthermore, through the combination of various classifiers in pairs and employing a voting ensemble approach, insights into performance metrics are gained. For instance, the combination of KNN and SVC yields a minimum accuracy of 97.471%, while the combination of AdaBoostClassifier and GradientBoostingClassifier achieves the highest accuracy of 99.205%.

3. Methodology

Under this section, the methods used for the execution of the study and implementation of the algorithms have been discussed. The diagram below shows the flowchart of the methodology.

Fig. 1 explains the methodology that has been used in the project. The study has used a dataset to train the cosine similarity model which is used for recommending movies. Then using another dataset, Naïve Bayes (NB) and Support Vector Machine (SVM) Classifier for Sentiment Analysis has been trained. Now a movie name is taken as an input and sent to the movie recommendation model to predict similar movies. Through web scraping from the IMDB site, the reviews of that movie are obtained and sent to the Sentiment Analysis model for classifying the reviews as positive or negative. We mainly using 10 classifiers 'Bagging', 'Random Forest', 'Decision Tree', 'Perceptron', 'SVC','KNN', 'Logistic Regression',

'ExtraTreesClassifier','GradientBoostingClassifier'and 'AdaBoostClassifier'

And we combine two classifiers and three classifiers.

1 Dataset

Three datasets have been used for study. 2 of them are for Movie Recommendation and 1 is for Sentiment Analysis. The ones used for recommendation are 'tmdb_5000_movies.csv', 'tbmd_5000_credits.csv' and the one used for sentiment analysis is 'reviews.txt'. The 2 datasets usedin movie recommendation are then merged to form a single data set and the columns kept under it are 'movie_id', 'title' and 'tags'.

The reviews data set has only 2 columns, one for the 'reviews' and other for the 'comments'. The positive comments have been labelled as 1 and the negative ones have been labelled as 0. There are 3943 positive comments and 2975 negative comments .

1 Data pre- processing

After merging the 2 datasets into a single dataset, only the essential columns such as 'movie_id', 'title'. 'overview', 'genres', 'keywords',' cast' and 'crew' are kept, rest are removed from the dataset .

Then using Abstract Syntax trees, the columns of 'genres', 'keywords', 'cast', 'crew' have been refined.

Furthermore, these columns have been combined under 'tags'. Then using the count vectorizer, the column 'tags' is tokenised. Tokenising means to divide the sentences into words. Here the pre-processing for Movie Recommendation comes to an end.

The pre-processing for Sentiment Analysis requires Natural Language Tool Kit (NLTK). The NLTK is a leading platform for building Python programs to work with human language data.

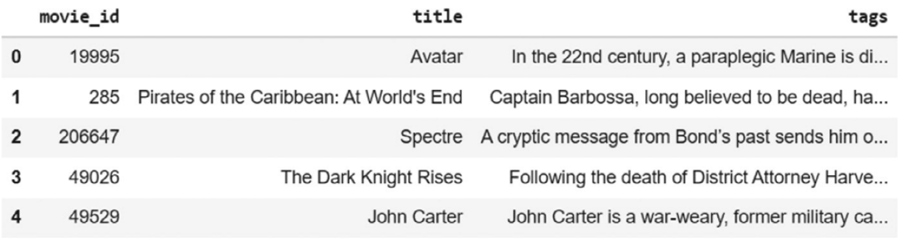
This is a standard python library used for natural language processing and computational linguistics. Using this library, the stop words are downloaded. Stop Words are the usually used words in any language. Examples of such words include 'a', 'an', 'the', 'if', 'or'. They are used in Text Mining and Natural Language Processing (NLP) to eliminate such words as they carry very little useful information. Then using the TfidfVectoriser, the column of Comments is tokenised.

Now the Data Pre-processing of the datasets ends here.

The Fig. 2 shows the dataset used for Movie Recommendation, which has the 'movie_id', 'title' and 'tags'.

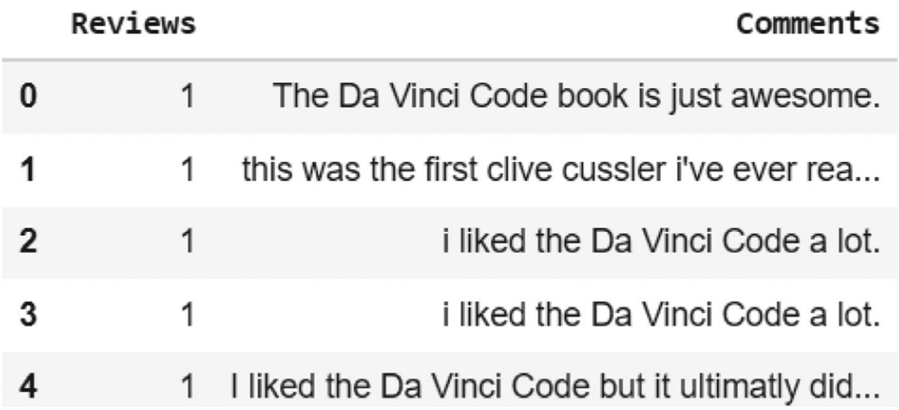
The Fig. 3 shows the dataset for sentiment analysis, has 2 columns, one for the 'comments' and other for the 'reviews'.

1 Data mining for movie recommendation



	movie_id	title	tags
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...
2	206647	Spectre	A cryptic message from Bond's past sends him o...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...
4	49529	John Carter	John Carter is a war-weary, former military ca...

Fig. 2. Final dataset being used for movie recommendation.



	Reviews	Comments
0	1	The Da Vinci Code book is just awesome.
1	1	this was the first clive cussler i've ever rea...
2	1	i liked the Da Vinci Code a lot.
3	1	i liked the Da Vinci Code a lot.
4	1	I liked the Da Vinci Code but it ultimatly did...

Fig. 3. Dataset for sentiment analysis.

Using the sklearn library in Python, the Cosine Similarity algorithm is used.

After the user is prompted to enter a movie, the algorithm provides 5 other movies like the one used as an input by the user.

In cosine similarity, vectors are taken as the data objects in data sets, when defined in a product space, the similarity is figured out. The smaller this distance, the higher the similarity, but the larger the distance, the lower the similarity. Cosine similarity is a measure that helps to find out how similar data objects are, regardless of size. Mathematically, it is the cosine of the angle between two vectors projected in a multi-dimensional space [9].

$$Cos\,\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\,\|\vec{b}\|} = \frac{\sum_{1}^{n} a_i b_i}{\sqrt{\sum_{1}^{n} a_i^2}\sqrt{\sum_{1}^{n} b_i^2}} \tag{1}$$

The angle between two vectors determines its direction and is measured in 'θ'. This angle θ can be calculated by using Eq. (1).

When θ = 0 °, the 'x' and' y' vectors overlap and prove to be similar.

When θ = 90 °, the 'x' and' y' vectors are therefore dissimilar.

1 Machine learning for sentiment analysis

As discussed earlier, 2 algorithms have been used, SVC and NB. The data set is split into testing set and training set, the testing set is 0.20 and training size is 0.80.

After this the 2 models are fitted. To increase the accuracy of both the models, hyperparameter tuning is applied on both the models.

SVC is a supervised algorithm in the machine learning domain used for both classification and regression. It classifies the info points by finding a hyperplane in an N-dimensional space [12]. The hyperplane is simply a line if the amount of input features is 2, however it's 2-D hyperplane if the amount of input features is three.

Radial Basis Function Kernel was used in the model which is a type of Non-linear SVM [13].

It is referred to as RBF kernel. Metric squared Euclidean distance is used for distance. It is used to draw completely non-linear hyperplanes.

$$K\left(X, X'\right) = exp\left(-\frac{\|X - X'\|^2}{2\sigma^2}\right) \tag{2}$$

Eq. (2) calculates function for two points X and X' computes the similarity or how close they are to each other.

'σ' is the variance and the hyperparameter. ||X – X'|| is the Euclidean (L_2-norm) Distance between two points X and X'.

The NB Algorithm uses conditional probability to classify the given data set. Bayes theorem is used for the computation and used class levels represented as feature values or vectors of predictors for classification [12].

$$P\,(A|B) = \frac{P\,(B|A)P\,(A)}{P\,(B)} \tag{3}$$

Eq. (3) calculates the conditional probability of event A such that B has already occurred, and this is used to for calculation in the NB classifier.

Types of models under NB algorithm

1 Gaussian,
2 Multinomial,
3 Bernoulli.

The proposed system uses the multinomial NB model, which predicts the badge of a text such as a piece of email or newspaper article. The probability of each badge is calculated for a given sample and then badge with the highest probability is given as output.

Since this algorithm is mainly used for natural language processing and text data analysis it was a perfect choice for sentiment analysis of movie reviews

4. Results

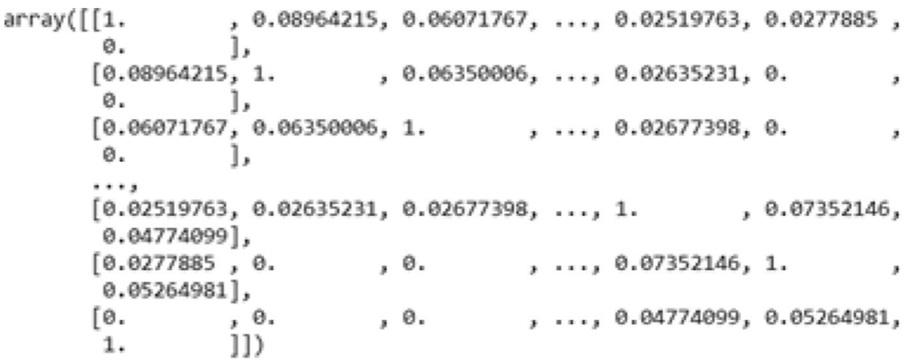
Using the Cosine Similarity algorithm, prediction of movies like the ones input by the user can be made.

The Fig. 4 shows the Cosine Similarity matrix of the values in the dataset.

The Fig. 5 shows that a negative comment has been classified as 0 by the NB classifier.

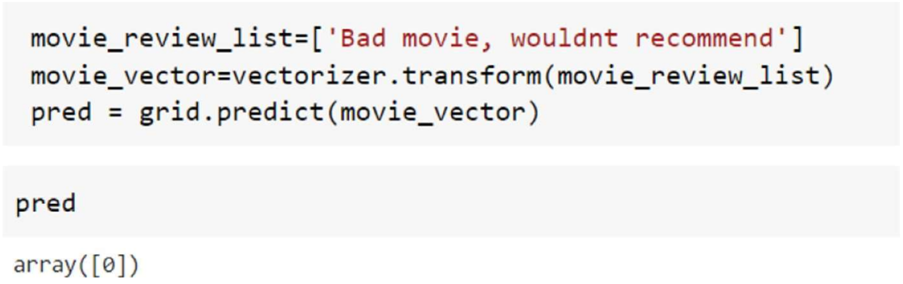
The Fig. 6 shows that a negative comment is classified as 0 by the SVC.

An integral part of the study was comparison between the 2 algorithms as well.



```
array([[1.        , 0.08964215, 0.06071767, ..., 0.02519763, 0.0277885 ,
        0.        ],
       [0.08964215, 1.        , 0.06350006, ..., 0.02635231, 0.        ,
        0.        ],
       [0.06071767, 0.06350006, 1.        , ..., 0.02677398, 0.        ,
        0.        ],
       ...,
       [0.02519763, 0.02635231, 0.02677398, ..., 1.        , 0.07352146,
        0.04774099],
       [0.0277885 , 0.        , 0.        , ..., 0.07352146, 1.        ,
        0.05264981],
       [0.        , 0.        , 0.        , ..., 0.04774099, 0.05264981,
        1.        ]])
```

Fig. 4. Value of array after fitting cosine similarity on the dataset.

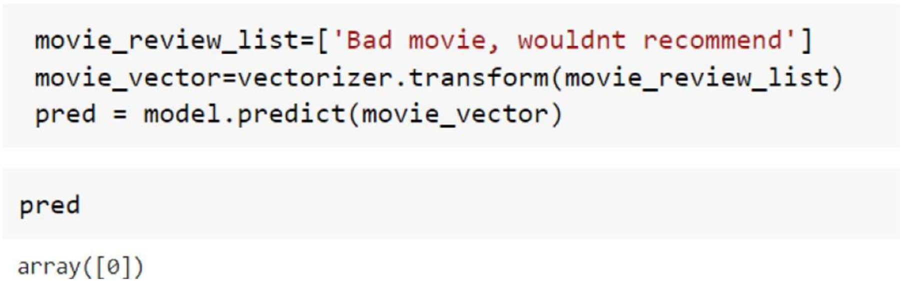


```
movie_review_list=['Bad movie, wouldnt recommend']
movie_vector=vectorizer.transform(movie_review_list)
pred = grid.predict(movie_vector)
```

```
pred
```

```
array([0])
```

Fig. 5. Output for NB algorithm.



```
movie_review_list=['Bad movie, wouldnt recommend']
movie_vector=vectorizer.transform(movie_review_list)
pred = model.predict(movie_vector)
```

```
pred
```

```
array([0])
```

Fig. 6. Output by SVC.

Table 1
Comparison of models

Algorithm	Accuracy	Precision	Recall	AUC
Naives Bayes (Proposed)	0.9733	0.96940	0.9850	0.970
SVM (Proposed)	0.9863	0.98278	0.9937	0.984
Bernoulli's Naive Bayes(Existing) [11]	0.875	0.884	0.8633	0.8735
Multinomial NB(Existing) [11]	0.885	0.9294	0.8333	0.8787
SVM(Existing) [11]	0.8733	0.859	0.8933	0.8753
NB(Existing) [5]	0.8183	0.84	0.79	0.82
SVM(Existing) [5]	0.8745	0.87	0.88	0.88
Random Forest (Existing) [5]	0.9601	0.93	1.00	0.96
Stacked-LSTM(Existing) [14]	0.9365	0.94	0.94	-
Minimal-RNN(Existing) [14]	0.8564	0.86	0.86	-
CNN(Existing) [15]	0.8915	0.8259	0.8246	0.8253
LSTM(Existing) [15]	0.9550	0.9087	0.8228	0.8636

The Table 1 shows the comparison between different Accuracy, Precision, Recall and AUC scores of the 2 proposed models and 3 existing models (Bernoulli's Naïve Bayes, Multinomial NB, SVM) studied in [11], 3 existing models (SVM, NB, Random Forest) studied in [5] 2 existing models (Stacked-LSTM, Minimal-RNN) studied in [14] and 2 existing models (CNN, LSTM) studied in [15].

Proposed SVM model is better than NB in all parameters.

The Fig. 7 shows the ROC Curves between the 2 algorithms.

One of the examples that taken for the study is the movie Spectre.

Through the Cosine Similarity algorithm, predictions of 5 other movies - Quantum of Solace, Never Say Never Again, Skyfall, Thunderball, From Russia with Love were made.

Fig. 8, shows the reviews about the movie - Spectre, entered by the user. Web scraping is used to get the taglines of the reviews. Web scraping is performed from the IMDB website and perform Sentiment Analysis on it using the NB and SVC algorithm.

After performing Sentiment Analysis, (Figs. 9, 11), one can tell whether it is a good movie or not. Algorithm 1: Clear the accuracy results obtained by using 10 classifiers and select the maximum accuracy.

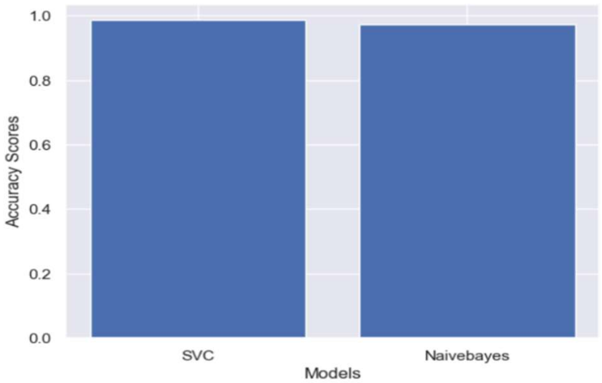
Algorithm 2: Determine the maximum accuracy achieved by combining two classifiers. Algorithm 3: Identify the maximum accuracy obtained by combining three classifiers.

So, the review 'Enjoyable installment in Bond series with lots of noisy action, thrills, emotion and spectacular scenes' gets assigned a value 1 which is the interpretation of a good review to the movie.

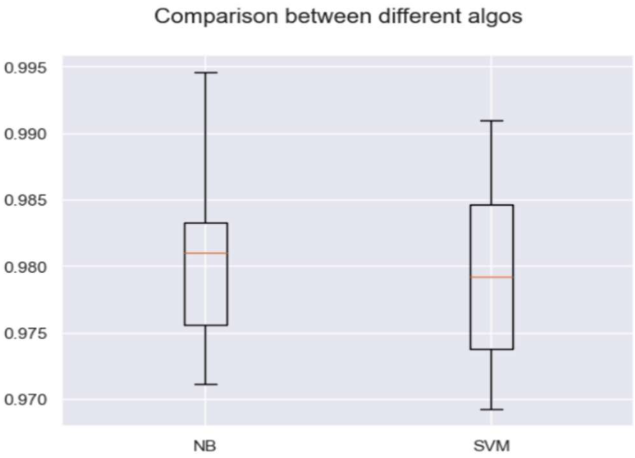
Another review 'the James Bond franchise should have ended decades ago' gets assigned a value 0 which is the interpretation of a bad review to the movie.

Comparison between the SVC and Naive Bayes (NB) algorithms in terms of accuracy shows that the accuracy of SVC is higher in all two figures (1,2).

1:



2:



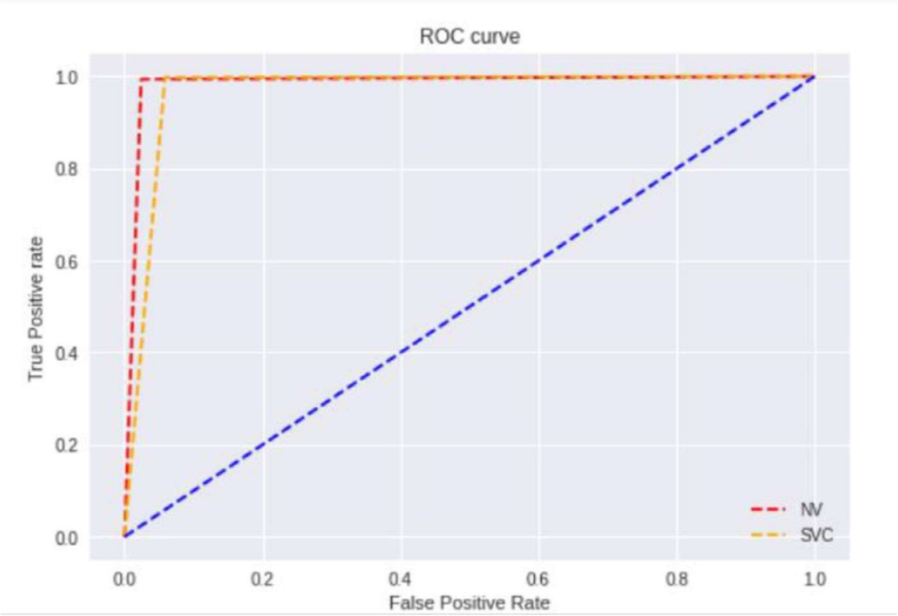
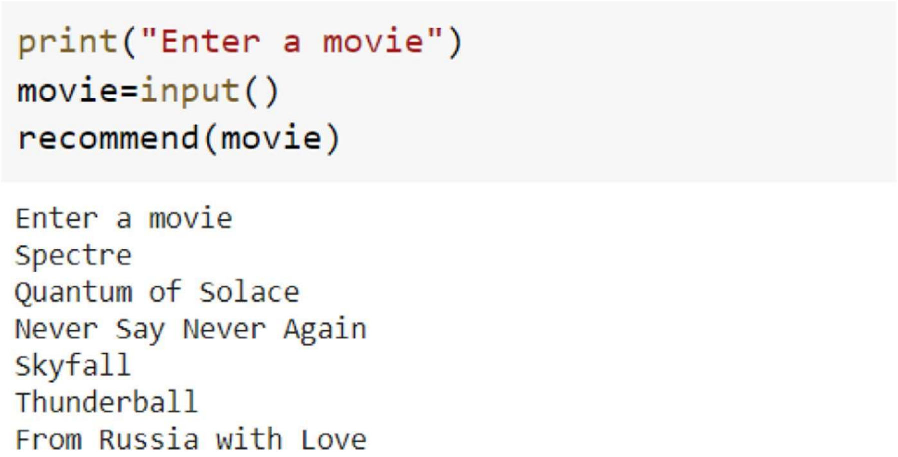


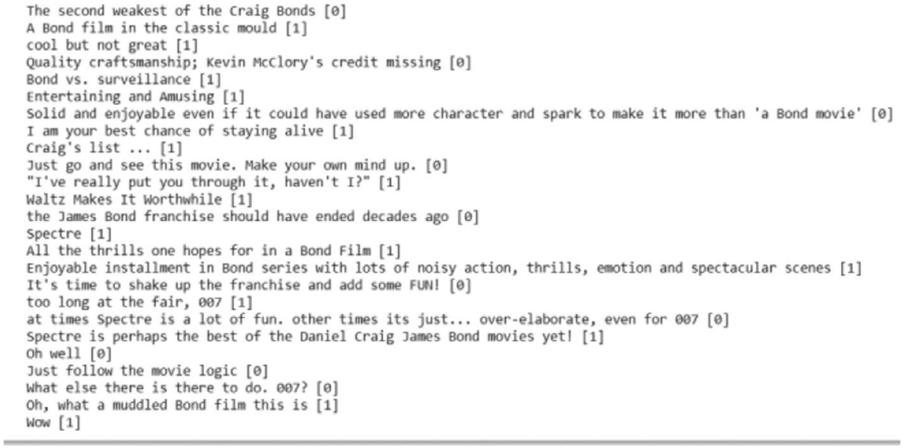
Fig. 7. ROC Curves.



```
print("Enter a movie")
movie=input()
recommend(movie)
```

```
Enter a movie
Spectre
Quantum of Solace
Never Say Never Again
Skyfall
Thunderball
From Russia with Love
```

Fig. 8. Prediction of movies using the cosine similarity algorithm.

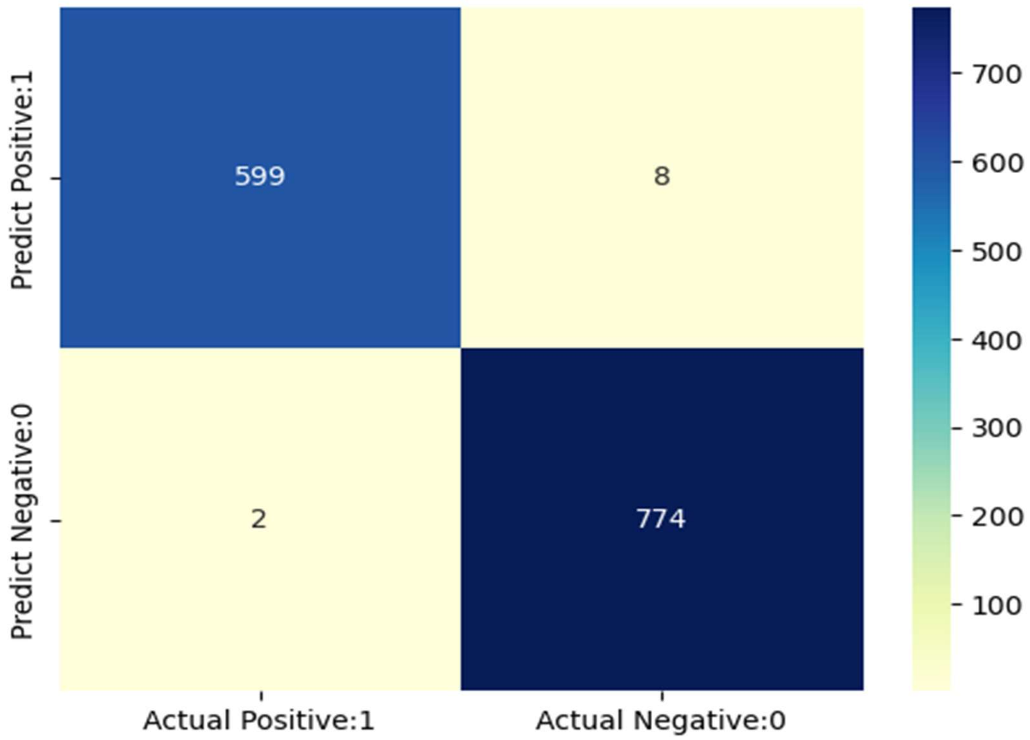


```
The second weakest of the Craig Bonds [0]
A Bond film in the classic mould [1]
cool but not great [1]
Quality craftsmanship; Kevin McClory's credit missing [0]
Bond vs. surveillance [1]
Entertaining and Amusing [1]
Solid and enjoyable even if it could have used more character and spark to make it more than 'a Bond movie' [0]
I am your best chance of staying alive [1]
Craig's list ... [1]
Just go and see this movie. Make your own mind up. [0]
"I've really put you through it, haven't I?" [1]
Waltz Makes It Worthwhile [1]
the James Bond franchise should have ended decades ago [0]
Spectre [1]
All the thrills one hopes for in a Bond Film [1]
Enjoyable installment in Bond series with lots of noisy action, thrills, emotion and spectacular scenes [1]
It's time to shake up the franchise and add some FUN! [0]
too long at the fair, 007 [1]
at times Spectre is a lot of fun. other times its just... over-elaborate, even for 007 [0]
Spectre is perhaps the best of the Daniel Craig James Bond movies yet! [1]
Oh well [0]
Just follow the movie logic [0]
What else there is there to do. 007? [0]
Oh, what a muddled Bond film this is [1]
Wow [1]
```

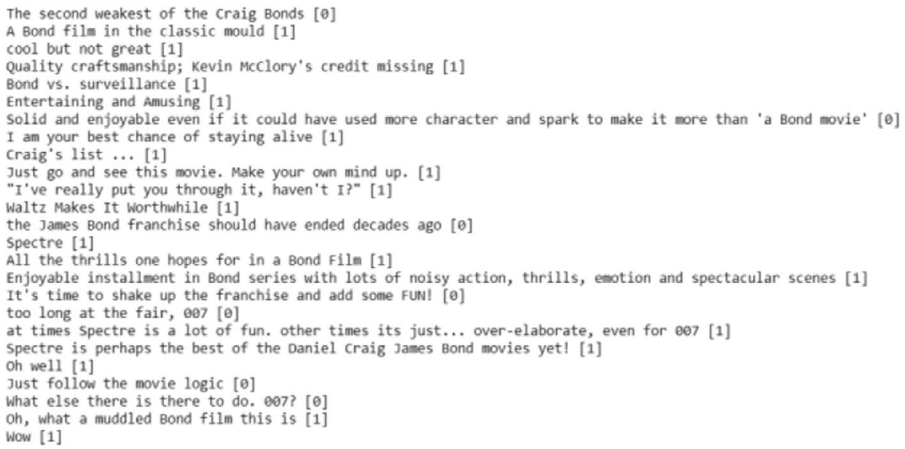
Fig. 9. Sentiment analysis on reviews using NB algorithm.

Visualizing Confusion

```
<Axes: >
```



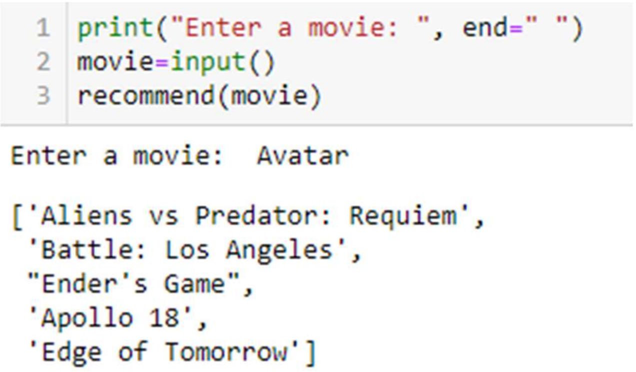
	Actual Positive:1	Actual Negative:0
Predict Positive:1	599	8
Predict Negative:0	2	774



```
The second weakest of the Craig Bonds [0]
A Bond film in the classic mould [1]
cool but not great [1]
Quality craftsmanship; Kevin McClory's credit missing [1]
Bond vs. surveillance [1]
Entertaining and Amusing [1]
Solid and enjoyable even if it could have used more character and spark to make it more than 'a Bond movie' [0]
I am your best chance of staying alive [1]
Craig's list ... [1]
Just go and see this movie. Make your own mind up. [1]
"I've really put you through it, haven't I?" [1]
Waltz Makes It Worthwhile [1]
the James Bond franchise should have ended decades ago [0]
Spectre [1]
All the thrills one hopes for in a Bond Film [1]
Enjoyable installment in Bond series with lots of noisy action, thrills, emotion and spectacular scenes [1]
It's time to shake up the franchise and add some FUN! [0]
too long at the fair, 007 [0]
at times Spectre is a lot of fun. other times its just... over-elaborate, even for 007 [1]
Spectre is perhaps the best of the Daniel Craig James Bond movies yet! [1]
Oh well [1]
Just follow the movie logic [0]
What else there is there to do. 007? [0]
Oh, what a muddled Bond film this is [1]
Wow [1]
```

Fig. 11. Sentiment analysis on reviews using SVC.

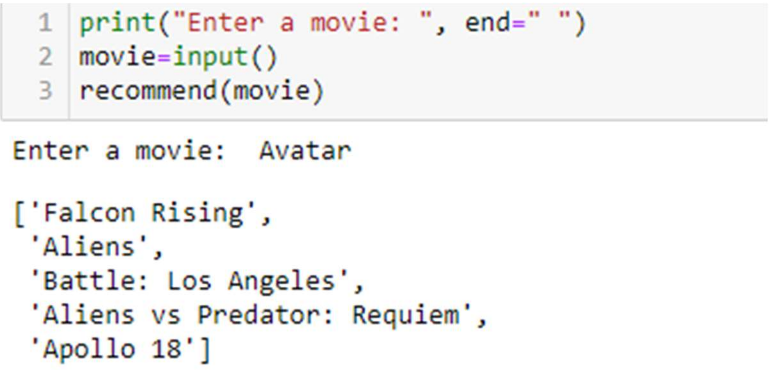
Fig12: **Results before the modification:**



```
print("Enter a movie: ", end=" ")
movie=input()
recommend(movie)
```

```
Enter a movie:  Avatar

['Aliens vs Predator: Requiem',
 'Battle: Los Angeles',
 "Ender's Game",
 'Apollo 18',
 'Edge of Tomorrow']
```

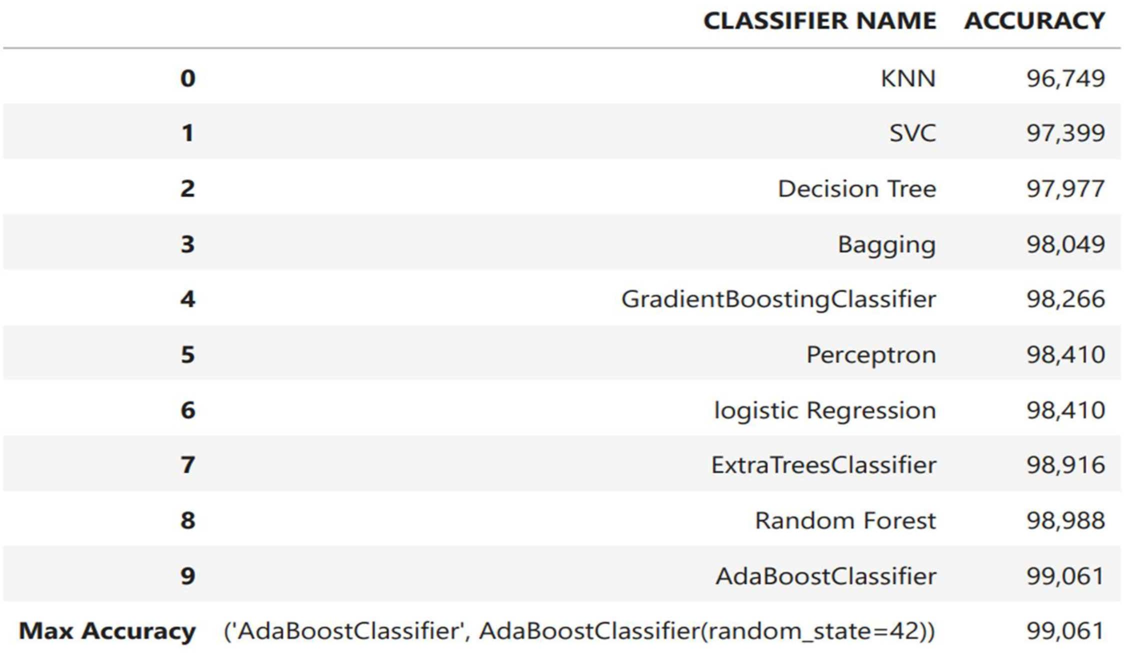
Fig13: **Results after the modification:**



```
print("Enter a movie: ", end=" ")
movie=input()
recommend(movie)
```

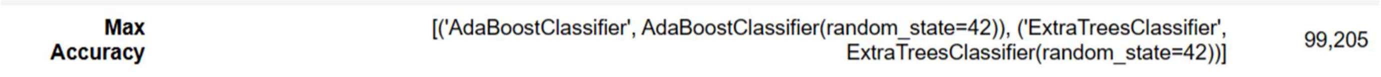
```
Enter a movie:  Avatar

['Falcon Rising',
 'Aliens',
 'Battle: Los Angeles',
 'Aliens vs Predator: Requiem',
 'Apollo 18']
```

Algorithm 1: Apply ten classifiers.



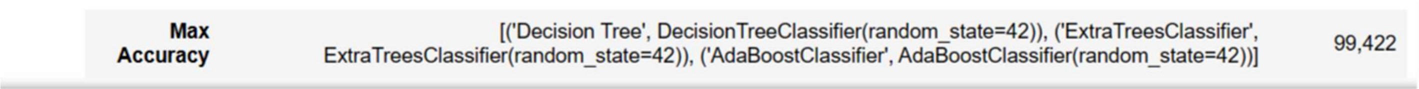
	CLASSIFIER NAME	ACCURACY
0	KNN	96,749
1	SVC	97,399
2	Decision Tree	97,977
3	Bagging	98,049
4	GradientBoostingClassifier	98,266
5	Perceptron	98,410
6	logistic Regression	98,410
7	ExtraTreesClassifier	98,916
8	Random Forest	98,988
9	AdaBoostClassifier	99,061
Max Accuracy	('AdaBoostClassifier', AdaBoostClassifier(random_state=42))	99,061

Algorithm 2: combine two classifiers.



Max Accuracy	[('AdaBoostClassifier', AdaBoostClassifier(random_state=42)), ('ExtraTreesClassifier', ExtraTreesClassifier(random_state=42))]	99,205

Algorithm 3: combine three classifiers.



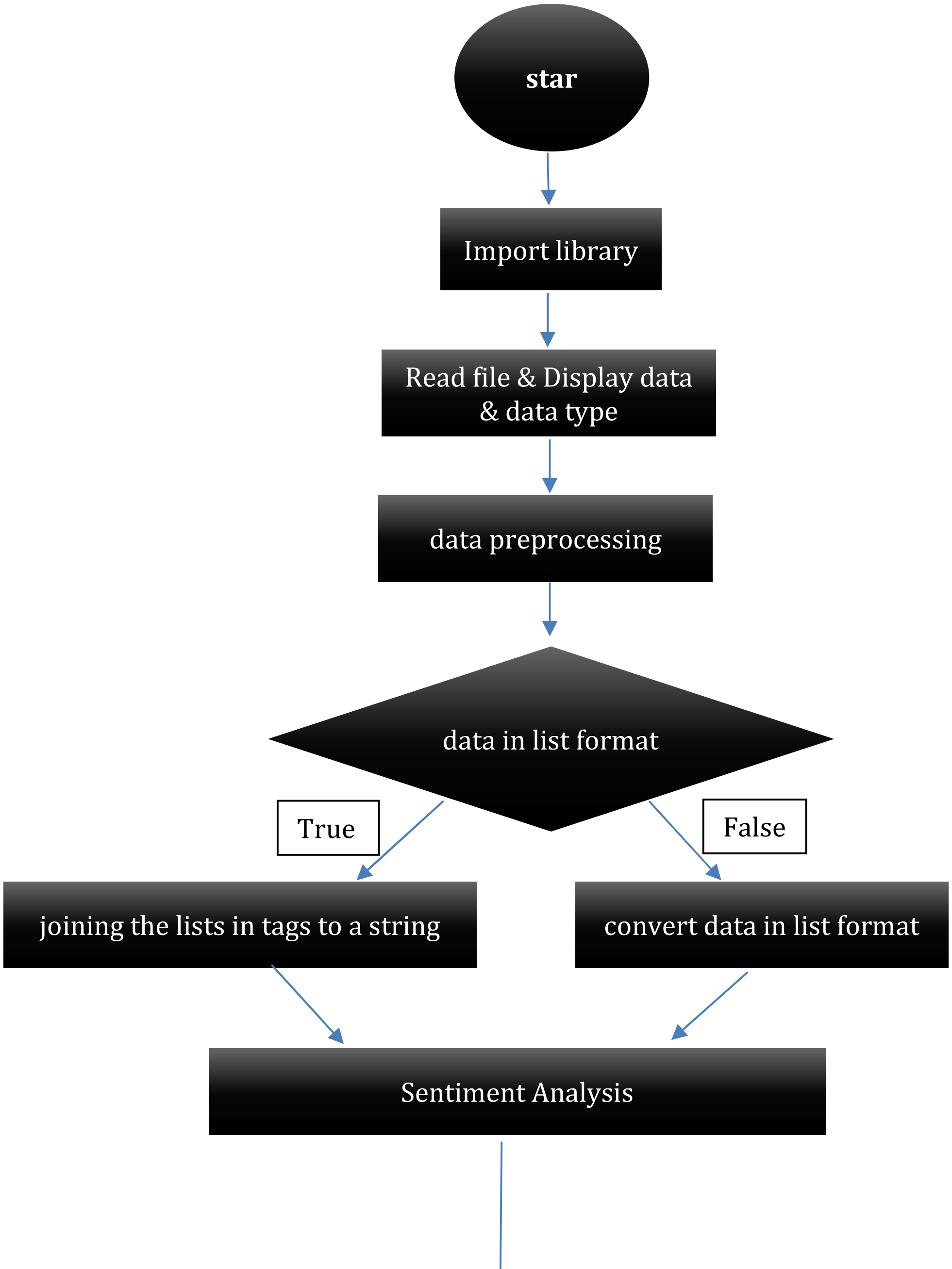
Max Accuracy	[('Decision Tree', DecisionTreeClassifier(random_state=42)), ('ExtraTreesClassifier', ExtraTreesClassifier(random_state=42)), ('AdaBoostClassifier', AdaBoostClassifier(random_state=42))]	99,422

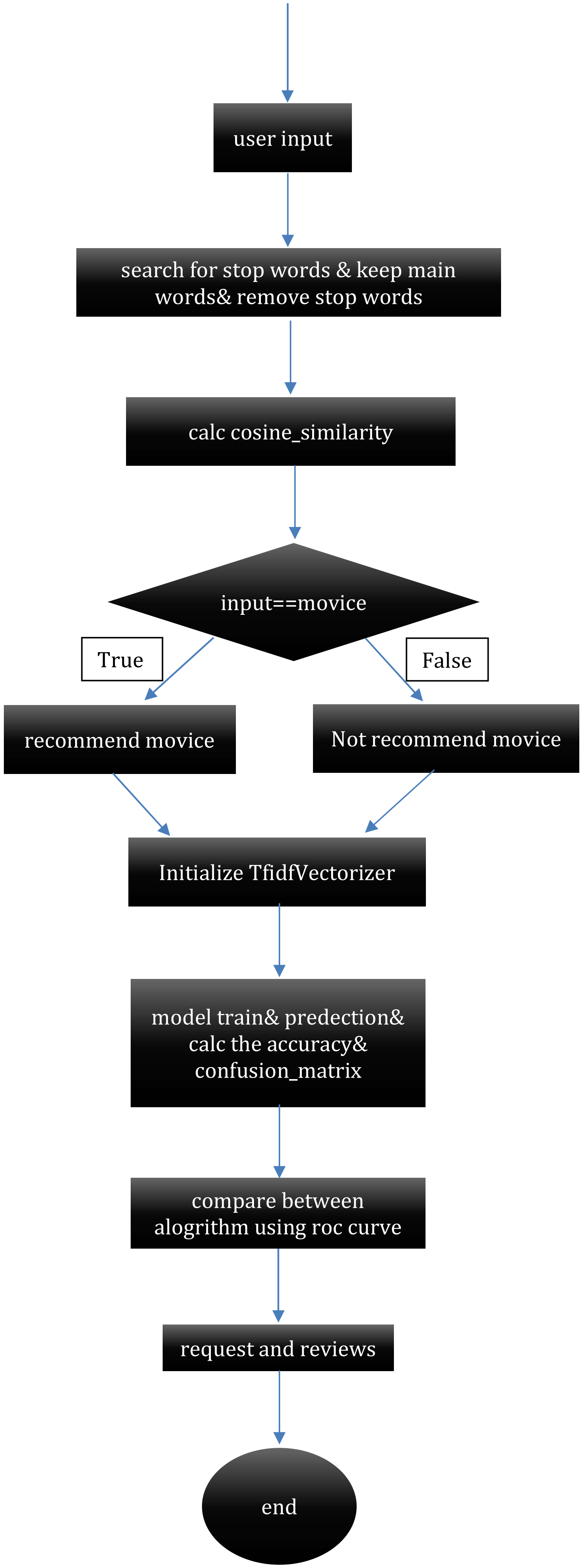
5. Pseudocode

```
Import necessary libraries
Load movie data from CSV files
Merge two datasets based on title column
Select essential columns and drop unnecessary ones
Define function to extract names from JSON objects
Apply this function to genres, keywords, and cast columns
Drop NA values in the dataset
Define function to fetch director name from crew column
Apply this function to crew column
Process overview column by splitting into individual words
Combine all columns into a single tags column
Create new dataset with only essential columns
Transform tags column into vector format using TF-IDF vectorizer
```

```
Define recommendation system function
Calculate similarity between movies using cosine similarity
Return top 5 recommended movies for given input movie
Load reviews data from text file
Preprocess reviews data by applying TF-IDF vectorizer
Perform grid search to find best parameters for XGBoost classifier
Train XGBoost classifier on preprocessed reviews data
Evaluate model performance using accuracy, F1 score, precision, and recall
Draw ROC curve and confusion matrix for model evaluation
Define pipeline for grid search and model training
Perform grid search to find best parameters for SVC classifier
Train SVC classifier on preprocessed reviews data
Evaluate model performance using accuracy, F1 score, precision, and recall
Draw ROC curve and confusion matrix for model evaluation
Compare performance of XGBoost and SVC classifiers using cross-validation
Draw box plot to compare algorithms
Define function to get IMDb reviews for given movie title
Fetch IMDb reviews using IMDbPY library
Preprocess reviews data by applying TF-IDF vectorizer
Predict review sentiment using trained models
Print results
```

6. Flowchart


star
Import library
Read file & Display data & data type
data preprocessing
data in list format
True
False
joining the lists in tags to a string
convert data in list format
Sentiment Analysis


user input
search for stop words & keep main words& remove stop words
calc cosine_similarity
input==movice
True
False
recommend movice
Not recommend movice
Initialize TfidfVectorizer
model train& predection& calc the accuracy& confusion_matrix
compare between alogrithm using roc curve
request and reviews
end

7. Conclusion

This paper presents a comprehensive analysis of two key components: the Movie Recommendation System and Sentiment Analysis. Through rigorous testing and evaluation, important insights and conclusions have been drawn. The Movie Recommendation System utilizes the Cosine Similarity algorithm to suggest relevant movies based on various factors such as genre, overview, cast, and ratings. Despite multiple tests, Cosine Similarity consistently delivers accurate movie recommendations, showcasing its reliability. In the realm of Sentiment Analysis, the study employs two algorithms, Naïve Bayes (NB) and Support Vector Machine (SVC), to classify reviews as positive or negative. Given the diverse nature of reviews, determining the optimal algorithm for classification is crucial. Experimental results reveal that SVC demonstrates slightly better accuracy compared to NB. Several avenues for future research are identified: Enhancing the accuracy of Sentiment Analysis to effectively classify sarcastic or ironic reviews. Extending Sentiment Analysis to reviews in languages other than English. Tailoring movie recommendations based on users' preferences, including cast, genre, and release year. Despite its accuracy, the system has limitations. Failure to recommend movies occurs when the user's input does not match the dataset format. Additionally, linguistic barriers restrict Sentiment Analysis to English reviews only. Moreover, misclassification may occur in the presence of sarcastic or ironic reviews. Furthermore, the study explores the utilization of 10 classifiers, with K-Nearest Neighbors (KNN) achieving the lowest accuracy of 96.749%, while AdaBoostClassifier demonstrates the highest accuracy of 99.061%. Employing voting ensembles with paired classifiers reveals insightful performance metrics, with the combination of KNN and SVC yielding a minimum accuracy of 97.471%, and AdaBoostClassifier and GradientBoostingClassifier achieving the highest accuracy of 99.205%. Additionally, utilizing three classifiers, 'Decision Tree', 'ExtraTreesClassifier', and 'AdaBoostClassifier', showcases remarkable performance, with AdaBoostClassifier achieving a maximum accuracy of 99.422%. These findings underscore the importance of recommending the highest accuracy classifiers for user utilization, enhancing overall system efficiency.

8. References

[1] N. Nassar, A. Jafar, Y. Rahhal, A novel deep multi-criteria collaborative filtering model for recommendation system, Knowl. Based Syst. 187 (2020) 104811.

[2] A. Beheshti, S. Yakhchi, S. Mousaeirad, S.M. Ghafari, S.R. Goluguri, M.A. Edrisi, Towards cognitive recommender systems, Algorithms 13 (8) (2020) 176.

[3] S. Sharma, V. Rana, M. Malhotra, Automatic recommendation system based on hybrid filtering algorithm, Educ. Inf. Technol. 27 (2021) 1–16.

[4] S.R.S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, B. Venkatesh, Content-based movie recommendation system using genre correlation, in: Smart Intelligent Computing and Applications, Springer, Singapore, 2019, pp. 391–397.

[5] M. Yasen, S. Tedmori, Movies reviews sentiment analysis and classification, in: Proceedings of the IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, JEEIT, 2019, pp. 860–865, doi:10.1109/JEEIT.2019.8717422.

[6] N. Rajput, S. Chauhan, Analysis of various sentiment analysis techniques, Int. J. Comput. Sci. Mob. Comput. 8 (2) (2019) 75–79.

[7] Z. Shaukat, A.A. Zulfiqar, C. Xiao, M. Azeem, T. Mahmood, Sentiment analysis on IMDB using lexicon and neural networks, SN Appl. Sci. 2 (2) (2020) 1–10.

[8] T. Widiyaningtyas, I. Hidayah, T.B. Adji, User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system, J. Big Data 8 (2021) 52.

[9] R.H. Singh, S. Maurya, T. Tripathi, T. Narula, G. Srivastav, Movie recommendation system using cosine similarity and KNN, Int. J. Eng. Adv. Technol. (IJEAT) 9 (5) (2020) 2–3 ISSN: 2249 –8958VolumeIssueJune.

[10] S. Kumar, K. De, P.P. Roy, Movie recommendation system using sentiment analysis from microblogging data, IEEE Trans. Comput. Soc. Syst. 7 (4) (2020) 915–923.

[11] A. Rahman, M.S. Hossen, Sentiment analysis on movie review data using machine learning approach, in: Proceedings of the International Conference on Bangla Speechand Language Processing (ICBSLP), IEEE, 2019, pp. 1–4.

[12] S. Uddin, A. Khan, M.E. Hossain, M.A. Moni, Comparing different supervised machine learning algorithms for disease prediction, BMC Med. Inf. Decis. Mak. 19 (1) (2019) 1–16.

[13] S. Ghosh, A. Dasgupta, A. Swetapadma, A study on support vector machine based linear and non-linear pattern classification, in: Proceedings of the International Con-ference on Intelligent Sustainable Systems (ICISS), IEEE, 2019, pp. 24–28.

[14] K. Dashtipour, M. Gogate, A. Adeel, H. Larijani, A. Hussain, Sentiment analysis of Persian movie reviews using deep learning, Entropy 23 (5) (2021) 596.

[15] S. Soubraylu, R. Rajalakshmi, Hybrid convolutional bidirectional recurrent neural network based sentiment analysis on movie reviews, Comput. Intell. 37 (2) (2021) 735–757.