**ABSTRACT**

# 

# The fast growth of the entertainment industry has resulted in the great interest in reputable ways of making predictions with regard to the quality of movies both before and after the release. In this work, we call “Predicting Movie Success from Reviews and Ratings with Machine Learning Techniques” we take numbers in the ratings and text from user reviews to imported models that can help to tell if a film might be successful or not. Through borrowing ideas from feedback and visiting numbers such as ratings, we are designing a thoughtful system that helps film creators and viewers make better decisions. The maximum precision was obtained as being 98.18 % in logistic regression which signifies the best effort was done to determine the relationship between the input film reviews and rating of the movies. The equally astounding growth was with respect to SVM Algorithm that yielded 96.36%. Data has an effect on how much people like movies, and delivers a dependable, versatile and sensible course of action to gauge their success in the venture.

# 

# CHAPTER 1

# INTRODUCTION

## 1.1 PROJECT DESCRIPTION

## The movies have long been a big part of world culture, both as a source of entertainment and Knowledge. A lot of movies appear annually, although not all of them sell a lot of tickets. Many of the others are of poor quality, and not everyone who views them is pleased. Lots of times, peoples who make the movies and persons who offer them read things like what critics say what peoples think about polls or what happened with very much similar movies over in the past to predicate will be strong movies. But these ways of taking aren’t always true or helpful. As a online services such as IMDb and Rotten Tomatoes have become popular, and it is convenient for users to give ratings and reviews. Having a lot of information helps public use Machine Learning to discover patterns and make better assumptions about whether a movie will be successful it.

## What this project is primarily interested in is why it is hard to determine whether a film is going to be a success before or after it is released. you’ll find views and ratings are scattered across other sites – making it difficult to gather useful input the old fashioned way. A single bad review may not be so, but when multiple viewers start leaving feedback and ratings, it could hint towards a pattern. The goal of this work is to create a Machine Learning model they can number ratings and written reviews and guess whether the movie will be a smash hit or not.

## The project focusing on developing a model that is able to analysis IMDb and Kaggle data with 50,000 movie reviews and Kaggle best for star ratings. It uses NLP to process the text data. and finally uses machine learning techniques to classify movies. The process starts with cleaning up and structuring the raw data. it applied Term Frequency-Invers Document Frequency to determine major features and train & test two models Logistic Regression (LR) & Support Vector Machine (SVM). To choose the model that makes the best forecasts, their accuracy is compared. The system is not just for school work it is also supposed to be utilized in the real world to help the movie business make improved decisions based on data. Future developments could utilize social media data's, watching video feedback for sentiments, or more smart training models in order to increasing accuracy.

# CHAPTER 2

**LITERATURE SURVEY**

Sharma et al. developed a prophetic model was suggested that employed several an Machine learning algorithms have been used to look at the IMDb film data. The test indicated that Logistic Regression worked well. [1]. Support Vector performed very well, Authors are achieving accuracy close by 90% when determining what feelings users were hits or flops. They explaining that user opinions deeply affect how movies are rated overall, but they didn’t consider merging of the text from reviews with other types of data like budget, cast, or box office collections. This gap shows that we need models that combine both same and quantitative data to better predicting the success of the movie is attributed to the fact that it contains humor However, it isn't amusing in and of itself. Authors, implemented sentiment way analysis model with ml classifiers based on IMDb reviews. In their research they find that Random Forest and Naive Bayes has very good accuracy [2]. but with SVM performing better in most of cases, with approximate 89% accuracy in emotion prediction for the viewers. They said audience feedback is what people think of a movie, which helps analyzing how achieved a movie might be early on early. However, their work only used text information and didn't include numbers or other types of extra data, which means there's room for better methods that merging different kinds of data, want what we've done. R. Saputri et al. And we will now (hopefully) use AI model to interpret human speech and disign out which parts of Avatar 2 we want or are hating. They based their reporting off of IMDb ratings and Kaggle reviews. Both LSTM and GRU were employed in a deep-learning-based model in [3]. This model considered affective aspects including the characters and their emotions. But the model [4] had a technical problem. 10 A. M. Sarhan et al proposed the smart movie recommendation system which not only recommend similar movies, but also knows what people feel about them.It reads reviews, both positive and negative, with AI models, and subsequently utilizes feelings and emotions to recommend the best movies that suit your mood. Their team worked on a complicated deep learning AI can automatically detect the sentiment of comments people post about movies online. Peter Atandoh and his team this is so that they (machines) can easily grasp emotions and opinions. In order to do this, they came up with a new model called Positional Embedding, It is a kind of a word order aware account. They also put it together with a Multichannel CNN and an Attention-based BiLSTM [5]. Authors in [6] developed an intelligent model that can read and comprehend the emotions and sentiments posted in movie reviews. I Steinke and others found that the top model was SVM, and it was able to correctly classify whether reviews were positive or negative with 86% accuracy. They also compared the reviews individuals wrote during the 2020 pandemic to the reviews individuals wrote prior to the pandemic in 2019. Authors in [7] developed a model that completely identified the sentiment in movie reviews. They used various methods, and one of them, Naïve Bayes, worked the best, achieving an accuracy of 81%. P. Baid and others created an intelligent system that looks at reviews and automatically tells if the opinions are positive or negative. This model helps save people's time. Authors from [8] developed an intelligent AI model that analyzes movie reviews one sentence at a time. A. Timmaraju and others looked at the emotion in each word and used that to Fig.ure out the overall feelings expressed in the comments. Their model worked well and used hand-crafted features such as flexibility and precision. Authors in [9] created a model that looks at how words are repeated and what feelings or emotions those words carry. This helped them build a smarter model that can understand the context and emotions in movie reviews. H. M. Keerthi Kumar, et al. Authors created a hybrid model name using SVM with ME particularly one which is good for analysis as positive and negative comments, is quick at extracting best features, and offers better accuracy when handling big size of datasets. Authors in [10] created a unique system that looks at and sorts through many films review and ratings, deciding if each one is positive or negative and star ratings. Alif Kahan, et al. made short summaries by picking the most useful sentences to help viewers better understand the opinions being shared. and reviews. M. J. Awan and others in paper [11] created an advanced model named Collaborative Filtering and Alternating baseline Movie Recommendation Engine with the use of Collaborative Filtering. Least Squares. This version belongs to Apache Spark and is intended to be used with big data. The ALS model It has predicted the top-rated movies. Authors focused on top-rated movies and recommended using deep learning models or MMD in feature work or research. This system was even smarter. Authors in these papers looked at two models. One of them was Random Forest, and the other was XGBoost. Zahabiya Mhowwala and others in [12] worked on getting the highest accuracy. The XGBoost algorithm achieved the highest accuracy, which was 95%. Their work was expanded to cover more movies. Authors described where their work might be useful, such as in the entertainment sector and for investors. A. Sharma and U. Ghouse created a CNN-BiGRU deep learning model, which itself processes Hindi movies A natural language processor will be used in reviews. The current paper [13] fills this gap. It analysed the English, Hindi and Indian languages through processing techniques like TF-IDF. This is done by means of these techniques. It establishes whether the attitude of Hindi films is good or bad. In [14], authors developed the model. A SVM algorithm was used to make out whether a sentiment is either positive or negative. This The King Fisher method attained a level of 89 percent accuracy which is design than Naive Bayes. D. Subedi and others wanted to find out which method, SVM or Naïve Bayes, gave the best accuracy. Authors mentioned that The SVM compared better in accuracy and other factors used to evaluate performance. Writers by using the IMDb. as an environment, that has the data informing about the actors and films, a model was developed with a methodology called SVM toa. This is classifying movie comments and based ratings on sentiment. I. S. K. Wardhana and others were combining TF-IDF features with proper preprocessing steps. They recommended continuing with more development by using better models or ideas from deep learning. M. Sanwal and others authors are created cleverest system that suggest movies, redacting how popular they might become using film features such as Action, Comedy, et al. It is an intelligent system The first study was done by introducing phrases, which were taken as the temporal context. Researchers were dealing with such fundamental data as a genre of movies, acts and directors who took part in them, and the plot they follow. Authors in [16] used ratings and votes from similar movies as their sources. By using IMDb and TMDb data sets, they reached an accuracy of 96%. 61% accuracy is achieved when creating a new dataset with the highest possible accuracy. S. Sahu and others designed a This movie rating prediction system relies on both Traditional machine learning and deep learning methods to forecast the coming ratings of films, in light of information in the IMDb and Movie lenses datasets. Authors concluded in [17] that users can be grouped based on their non-average and average user ratings, matching how they rate films on IMDb.

# CHAPTER 3

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM AND LIMITATIONS**

In the present systems, based on user ratings and reviews alone is how movies are predicted to be successful. The models analyze text reviews using sentiment analysis and machine learning techniques like Naïve Bayes, Random Forest, or Support Vector Machine (SVM) are employed to identify whether a review is positive or negative. These techniques offer some useful knowledge, although they reveal only half of what the audience is expressing. Methods that use only numbers to grade something reveal how well-liked it is but fail to reveal the genuine thoughts and emotions people express in their reviews. The main problem of systems like this is the fact that they don't look at things completely. If you're just considering reviews or ratings alone, the predictions you'll make may not be very accurate and can be highly variable based on the data you're working with. Aside from that. The explored deep learning algorithms demand significant computer resources, making them unfitness for usage in basic or real-time scenarios.

Because there isn't one system that unites both organized data such as ratings and the more fluid information from reviews, the practices that are in place at the moment aren't nearly as efficient as they can be.

**3.2 PROPOSED SYSTEM**

The new system is made to resolve problems with the old ways by combining both user reviews and ratings into a machine learning method. This is more accurate in forecasting how well a movie will perform. This model makes more precise and comprehensive predictions using two types of data viewers' reviews and numerical data ratings, which is unlike previous systems that merely considered text or numbers.

The system employs a sequential method: reviews are cleaned and processed with NLP techniques to determine the emotions they convey, and the raw numbers from ratings are utilized as a portion of the data. These features together are trained and tested Using machine learning strategies like Logistic Regression and Support Vector Machine. The outcome is that the models perform highly well in prediction, where Logistic Regression achieves 98.18% and SVM achieves 96.36%, a merging two approach. The system connects written reviews with star ratings, which helps improve the accuracy, ability to handle more data, and real-world value for people involved in the movie industry, such as producers, distributors, and viewers.

**3.3 FEASIBILITY STUDY**

The feasibility study checks if the proposed system can be made and used in real-life situations. It looks at the technical, economic, and operational parts to make sure the project is useful and works well.

**3.3.1 Technical Feasibility**

The project is technically possible to accomplish because it employs technologies that are open-source and widely available Python, together with libraries like scikit-learn, pandas, NumPy, & NLP Tool Kit, offers a solid base for handling the use of Machine Learning and Processing of Natural Language projects. The system doesn't need powerful hardware because models like Logistic Regression and SVM are easy to compute and work well on regular computers.

**3.3.2 Economic Feasibility**

The development cost is low because the project uses open-source tools and free IMDb-style datasets. You don't need to use any paid software or maintain complex systems. This makes the system affordable for use in academic settings and also suitable for small industries to adopt.

**3.3.3 Operational Feasibility:**

The system is simple for use and can be operated smoothly by people like movie analysts, producers, and distributors. Users can enter movie information like reviews and ratings through an easy-to-use interface and get accurate predictions. Because The method has extremely accurate, it helps make better decisions and can easily fit into current work processes, which keeps everything running well.

The feasibility study indicates that the project is viable from a technical perspective, is budget-friendly in terms of expense, and can be implemented in a realistic manner. So, the system we're suggesting can be put into use and it will help predict how successful a movie will be in a correct and useful way.

**3.4 REQUIREMENT SPECIFICATIONS**

**3.4.1 Functional Requirements**

Functional demands specify the exact purpose and capability that the system must offer:

* The system needs to let people enter movie reviews and their ratings.
* The system needs to process reviews first Using methods from naturally occurring language Processing, which include breaking down converting the content into words, deleting notions that don't add much meaning, and simplifying the words in their simplest basic form
* The system needs to turn text reviews into numbers (like using TF-IDF or Bag-of-Words).
* The system must Use machine learning approaches likes logistic regression.and SVM to classify data.
* The system needs To generate an outcome that guesses if a movie will be a success or a failure.
* The system needs to show how accurate and how well the trained models are performing.

**3.4.2 Non-Functional Requirements**

Non-functional requirements define the quality attributes of the system:

* **Performance:** The system needs to give predictions fast and work well with bigger sets of data.
* **Scalability:** The system needs to create a result that predicts whether a production will be successful or a failure.
* **Usability:** The user interface must be simple to use and comprehend, particularly for those with less technical expertise.
* **Reliability:** The system needs to give dependable and correct results every time, no matter which set of data it uses.
* **Maintainability:** The code should be organized into separate parts and have clear explanations so it's easier to update and improve over time.
* **Portability:** The system needs to operate on a variety of OS, like Linux and Unix with very little setup needed.

# CHAPTER 4

**SYSTEM DESIGN**

**4.1 SYSTEM ARCHITECTURE DIAGRAM**

UI Layer (Reviews & Ratings Input Layer)

Preprocessing & NLP (Text Cleaning with TF-IDF)

Rating Dataset (using Numeric Features)

Feature Integration

(Combine feedback + Ratings)

Machine Learning Train & Test two Algorithms like Logistic Regression & SVM

Output (Predicating Flop & Hit, models Accuracy, Matrix)

Fig. 1 System Architecture for Hit-or-Flop Prediction Using User Reviews and Ratings.

Fig. 1 shows the system design uses both user reviews and ratings to Fig.ure out if an item will be successful or not. User inputs are gathered through the user interface, where they provide text-based reviews and numerical ratings. The reviews go through a process called preprocessing, which converts text into machine-understandable numerical features using the processing of natural languages approaches like textual cleaning and TF- IDF vectorization.

**4.2 DFD (DATA FLOW DIAGRAM)**

**Level 0**

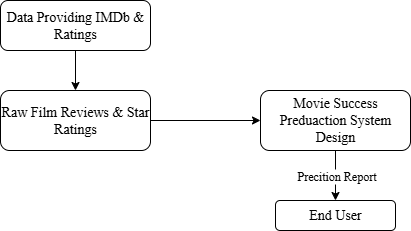
****

Fig. 2 Data Flow Diagram Level 0 for Movie Success Prediction System.

Fig. 2 shows the Level 0 A data flow diagram is also referred to as the Context Diagram, for the Movie Success Prediction System. The system gathers data sets from outside sources like IMDb and Kaggle. Users use the system by giving reviews and ratings, and the system uses these to do prediction and things how people imagination about the topic. The system gives users the results and insights, so they can see what the predictions are.

**Level 1**

Fig. 3 represents the Level 1 DFD where five key functional processes of the system are outlined. It reflects the stream of data coming in external sources into the pipeline, through significant stages such as preprocessing and feature engineering, and the stages of training the model and making predictions. This diagram illustrates well the flow of data between raw inputs, a stored model and a report that an end user can use to make decisions, and the primary data stores that are used to persist data.

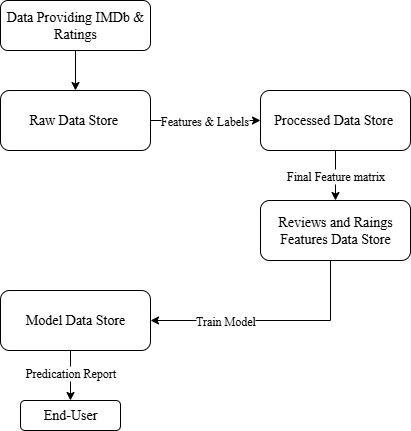


Fig. 3 Data Flow Diagram Level 1 for Movie Success Prediction System.

**Level 2**

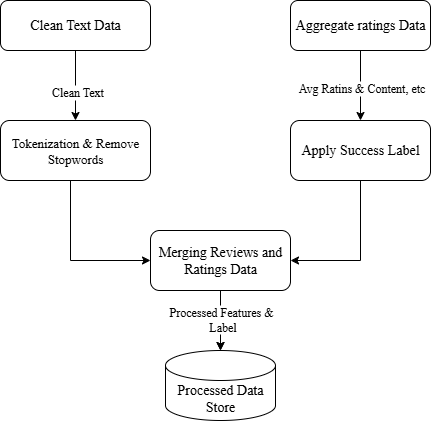


Fig. 4 Data Flow Diagram Level 2 for Movie Success Prediction System

.

Fig. 4 shows the Level 2 DFD provides an expanded break-down of the Preprocess & Transform Data activity in Level 1. It separates this one single process into five sequential sub processes: cleaning the raw text, tokenizing, aggregating numerical ratings, applying a business rule to create a success label, and collating these results into a single dataset. This close-up picture is the key to grasping the previously mentioned, petty, daily chores that have to be performed in order to transform the disorganized, raw information into the well-ordered, clean data, which can be utilized in the context of the training phase of machine learning algorithms' successes.

**4.3 FLOW CHART**

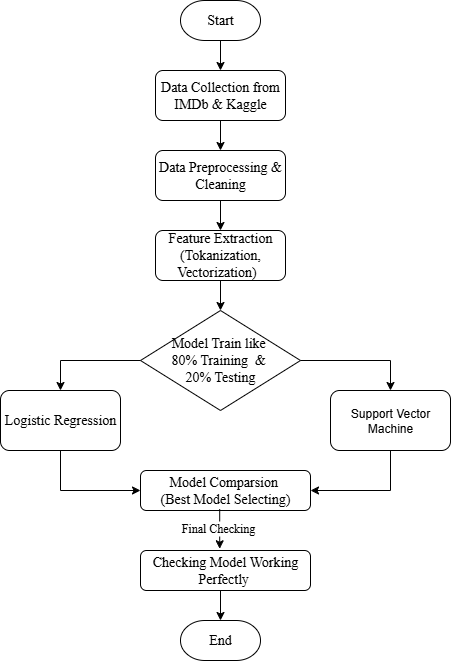
****

Fig. 5 Work Flow Diagram.

Fig. 5 represents the workflow for utilizing machine learning approaches to forecast the success of films.

* The process starts by gathering data from IMDb and Kaggle, which offer both movie reviews and their corresponding ratings.
* First, the collected data is filtered and prepared to get rid of unnecessary stuff, incomplete entries, and random errors.
* After that, a process called feature extraction, also known as vectorization, is used to convert text from reviews into numbers. This is done using techniques such as TF-IDF or word embeddings.
* The The algorithm gets trained using the dataset., and two algorithms are tested: Logistic Regression with an 80-20 split for training and testing, and SVM (Support Vector Machine) with a 70-30 split.
* Once the training is done, the models are tested and considered to determine which one performs best.
* Lastly, emotions are compared using the chosen model to assist ascertain if a film has a higher chance of succeeding or failing.
* The process ends once the results and comparisons are created.

# CHAPTER 5

**IMPLEMENTATION**

**5.1 TECHNOLOGIES USED**

The implementation of the Movie Success Prediction System involves several technologies that are data collection, preprocessing Methods, machine learning Models, and result visualization.

* **Programming Language:** ‘Python’ is used because it is very Simple, Flexible and best for Training and Testing ML Programs and analyzing data
* **Libraries & Frameworks:**
  + - **NumPy and Pandas:** This library is useful for handling Dataset, Preprocessing, Data Manipulation.
    - **Scikit-learn:** A library for applying ML Models like SVM and Logistic Regression measures.
    - **NLTK:** Library assisting in Processing Natural Language and Sentiment Analysis.
* **Data Sources:** ‘IMDb & Kaggle’ this both platforms used to collect Dataset Containing

movie reviews, film ratings and meta data for training and testing.

* **Visualization Tools:** ‘Matplotlib and Seaborn’this library helps to Data Visualization, Charts & Graphs, Performance metrics representation.
* **Environment & Tools:** ‘Jupyter Notebook / Google Colab’ this platform used to model development and experimentation.
* **Database:** ‘Xampp Server’ for storing result.

**5.2 FRONTEND AND BACKEND DETAILS**

* **Frontend**

The system's front part is made to offer an easy-to-use and engaging a means of user interaction. JavaScript, HTML, and CSS are used in its construction to make pages that work well on different devices and are easy for users to navigate. To improve the design, developers use tools like Bootstrap or Tailwind CSS.

The frontend's main purpose is to allow users to:

* Write movie reviews and give ratings.
* Interact with the system for predictions.
* Check the outcomes of the movie success forecast (Hit or Flop) along with the sentiment ratings.
* The front part of the app talks to the back part using Flask which helps data move smoothly and lets the app respond quickly to what users ask for.
* **Backend**

The back end is the primary component of the project, responsible for processing data, using machine learning models, and making predictions. It is built using Python, and it uses Flask or Django as the web framework.

The backend process involves:

* **Data Preprocessing:** Cleaning the raw text, breaking it into words, and removing common words that don't add much meaning.
* **Feature extraction**: involves converting text reviews into numerical vectors by applying sentiment evaluation techniques and TF-IDF.
* **Model Training:** using a review and ranking datasets train a machine learning strategies like Logistic Regression.
* **Prediction:** With the use of the most current data, the models that have been trained can make predictions about whether a film will be successful or not.
* **API Management:** Letting the frontend access endpoints to get predictions and show the results.

**5.3 CODE AND SCREENSHOTS**

**a. Frontend Code Snippet and Screen Shot**

<form id="predictionForm" class="space-y-4">

      <div>

        <label class="block text-gray-700 font-semibold">Movie Name</label>

        <input type="text" id="movieName" placeholder="Enter movie name"

          class="w-full p-2 border rounded-lg" required>

      </div>

      <div>

        <label class="block text-gray-700 font-semibold">IMDb Rating</label>

        <input type="number" step="0.1" id="rating" placeholder="Enter rating (1–10)"

          class="w-full p-2 border rounded-lg" required>

      </div>

      <div>

        <label class="block text-gray-700 font-semibold">Genre</label>

        <select id="genre" class="w-full p-2 border rounded-lg">

          <option>Action</option>

          <option>Drama</option>

          <option>Comedy</option>

          <option>Thriller</option>

          <option>Romance</option>

          <option>Horror</option>

          <option>Sci-Fi</option>

          <option>Animation</option>

        </select>

      </div>

      <div>

        <label class="block text-gray-700 font-semibold">Review</label>

        <textarea id="review" placeholder="Enter your review"

          class="w-full p-2 border rounded-lg"></textarea>

      </div>

      <button type="submit"

        class="w-full bg-blue-500 text-white py-2 px-4 rounded-lg hover:bg-blue-600">

        Predict

      </button>

    </form>

Fig. 6 the front end of It's a clean system and simple screen upon which individuals can input the title of the movie, its IMDb rating, whether it is a particular genre or not, and the opinions of others regarding it. HTML, JavaScript, CSS, and Tailwind's CSS are used in its design so that it is responsive and simple to use. Once the text is input, the user can click on the Predict button. This sends the data to the back end where the data is processed and utilized to make a prediction

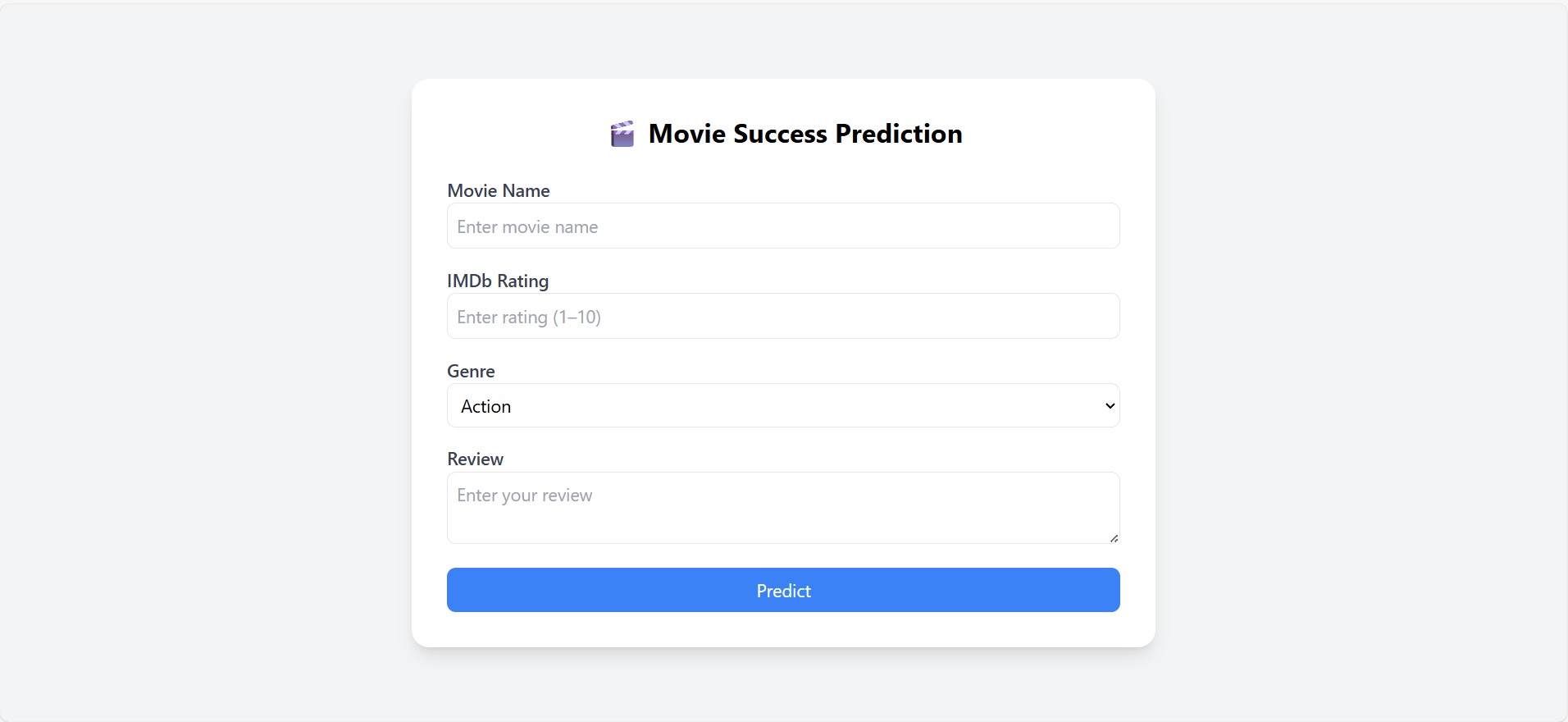


Fig. 6 Frontend End Screen Shot Movie Hits Prediction.

.

**b. Backend Code Snippet**

@app.route("/save", methods=["POST"])

def save():

    data = request.get\_json()

    movie\_name = data.get("movie\_name")

    rating = data.get("rating")

    review = data.get("review")

    prediction = data.get("prediction")

    try:

        conn = get\_db\_connection()

.

The above code opens the server page using Flask, a Python best framework, and it takes care of all the processing and prediction work. The output displays the Flask server operating in debug mode at <http://127.0.0.1:5000/>, which enables easily interaction between the front-end users and the machine learning model. When a user sends movie details from the app's main interface, the back part gets that info, works with it, and then gives back the predicted output.

**c. Database Code Snippet and Screen Shot**

CREATE TABLE Movies (

movie\_id integer primary key element,

title TEXT,

review TEXT,

rating REAL,

prediction TEXT

);

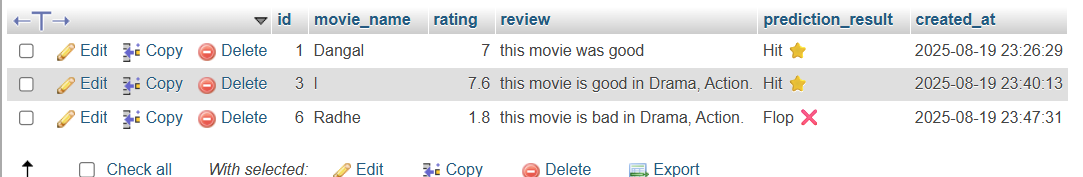


Fig. 7 Database Screen Shot and Storage Information.

Fig. 7 This is a snapshot of the project's database table where the outcome of the movie success prediction is stored. Every record contains data such as the title of the movie, the user rating, the text review, and the machine learning model-predicted outcome (Hit or Flop). For instance, the film Dangal, which has a rating of 7 and a review of "this movie was good," was forecasted to be a Hit. Radhe, with a low rating of 1.8 and a poor review, was forecasted to be a Flop. The table also monitors the time (created\_at) when each forecast was created, so you can actually monitor the system's outputs. This database storage is easy to track, verify, and learn the prediction system.

# CHAPTER 6

**TESTING**

**6.1 TESTING METHODS**

* + 1. **Unit Testing**

In this, testing of units was completed work t to verify that all small components function correctly. The input handling was tested to verify that movie names, ratings, and reviews are received and processed without any errors. The text preprocessing module was tested to verify that reviews are cleaned and converted to the correct format. The prediction module was tested to ensure the machine learning model accurately categorizes movies as either "Hit" or "Flop." Insertion in the database was also tested to ensure all records are correctly stored. Integrations testing a carried out to guarantee that every system's to many component together perform optimally.

**6.1.2 Integration Testing**

The link between the front component and the backend component was tested to confirm that user data is passed to the prediction model correctly. The step of moving the prediction output to the database was tested to confirm that it worked as expected. The database was added to the user interface, and the confirmation that the results saved in the database are shown of the viewers correctly was tested.

**6.1.3 System Testing**

To make sure the entire work runs smoothly, system testing occurred. The system was examined using various movie reviews and ratings to ensure that it gave the correct predictions. The database was tested to ensure all the results had corrected and clear data. The frontend was tested to ensure users received the correct prediction feedback. The system was tested thoroughly to ensure that it fulfilled all the specifications and functioned well in various test environments.

**6.2 TEST CASES AND RESULTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No. ID** | **Information** | **Inputs** | **Expected Outputs** | **Actual-Results** | **Status** |
| TC01 | Check data loading and preprocessing | Movie dataset (reviews & ratings) | Data set successfully loaded with the help of the missing values treated. | Data cleaned and loaded correctly | Pass |
| TC02 | Sentiment analysis on reviews | Raw text onions | Positive/Negative sentiment score | Sentiment labels generated as expected | Pass |
| TC03 | Feature extraction (TF-IDF / Bag of Words) | Preprocessed reviews | Numerical feature vectors | Features extracted successfully | Pass |
| TC04 | Logistic Regression model training | Training dataset (features + labels) | Logistic Regression train no errors | Model trained completed | Pass |
| TC05 | SVM model training | Training dataset (features + labels) | Support Vector Machine train no errors | Model trained completed | Pass |
| TC06 | Logistic Regression prediction | Test dataset | Analysis result is Hit or Flop | Predictions coming correctly | Pass |
| TC07 | SVM prediction | Test dataset | Predicted output are Hit or Flop | Predictions result correctly | Pass |

Table 1. Test Case Results.

# CHAPTER 7

# RESULTS AND DISCUSSION

**7.1 SYSTEM OUTPUTS**

**a. System Output Movie Flop Prediction:**

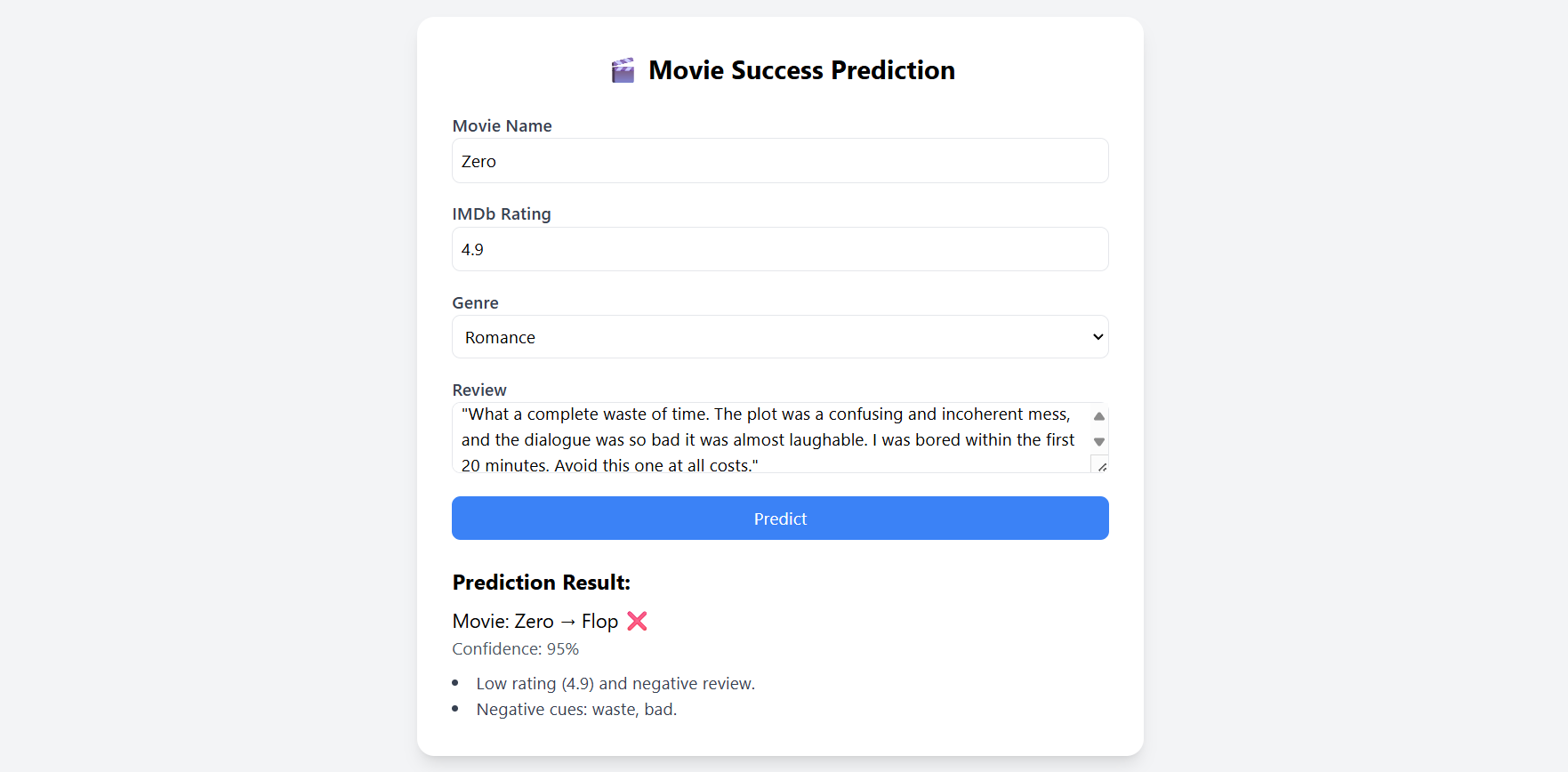
****

Fig. 8 System Output Movie Flop Prediction Through User Input.

**b. System Output Movie Data Store in Database**

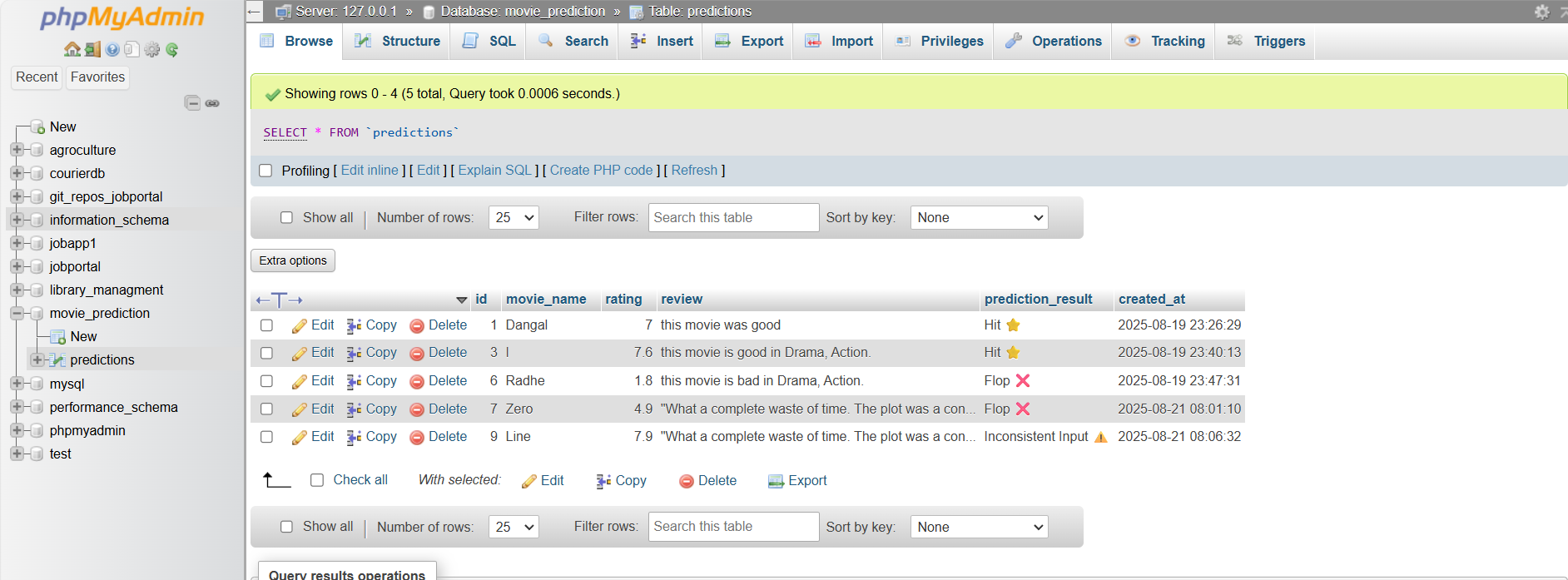
****

Fig. 9 All Outputs Store in MySQL Xampp Database.

In fig. 8 movie flop prediction is shown. In fig. 9 database is storing the all outputs is shown. The system's outputs clearly show that It can handle a variety of the circumstances when predicting how successful a movie will be. When the inputs include high ratings and good reviews, the system expects the movie to be a hit. In contrast, for movies that have low ratings and bad reviews, the system correctly identifies them as a flop. If there are conflicting inputs, like a high rating but a negative review, the system smartly labels the result as Inconsistent, helping the user know they need to check their inputs again. The different results shown in the images show how strong and dependable the model we created is.

**7.2 PERFORMANCE EVALUATION**

**a. Balanced Sentiment Distribution in Movie Review**s:

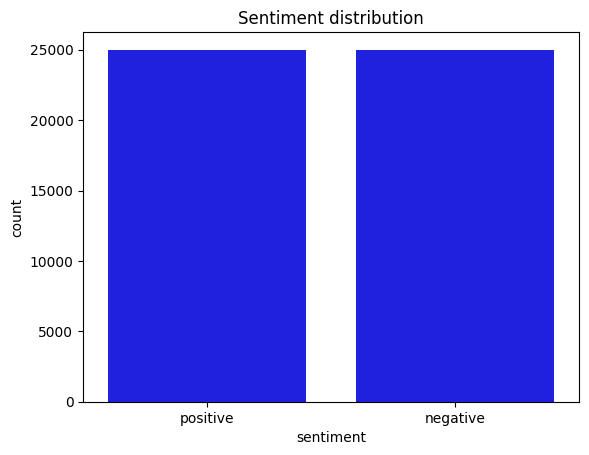
****

Fig. 10 Balanced Sentiment Distribution in Movie Reviews

Fig. 10 graph displays how the emotions in the dataset are spread out. First distribute the 25000 positive reviews and next negative reviews same as 25000. This graph explains the IMDb-films- dataset broken into two distinct parts such as Positive and Negative. This data set helps to achieves the best an accuracy using this Machine Learning algorithms.

**b. Top Ten Average Rating in Dataset**

Fig. 11 graph denotes the highest 10 average ratings in addition to tones noted in the information. The one that most frequent of these are the ratings 7.1 and 6.6 which each has 11 occurrences respectively meaning that it is a substantive number of movies that rate in the aforementioned range. Paying attention to the details, 9.0/7.7/5.7/7.8/9.3 are all the scores, which appear 10 times and proves that these scores are also very frequent in the dataset. In the meantime, the ratings 8.1, 8.0, and 8.5 are somewhat less commonly seen, 9 of their occurrences that make it to the top 10 list of ratings. In sum, the distribution indicates that majority of movies in the dataset are concentrated along ratings between 7.0 and 9.0, with the general trend being moderately high to high ratings

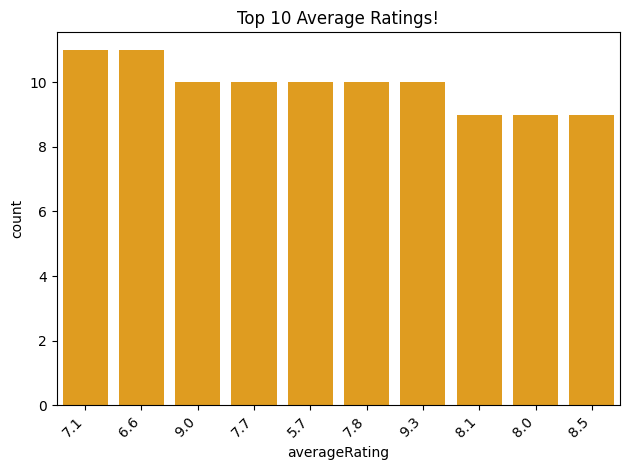
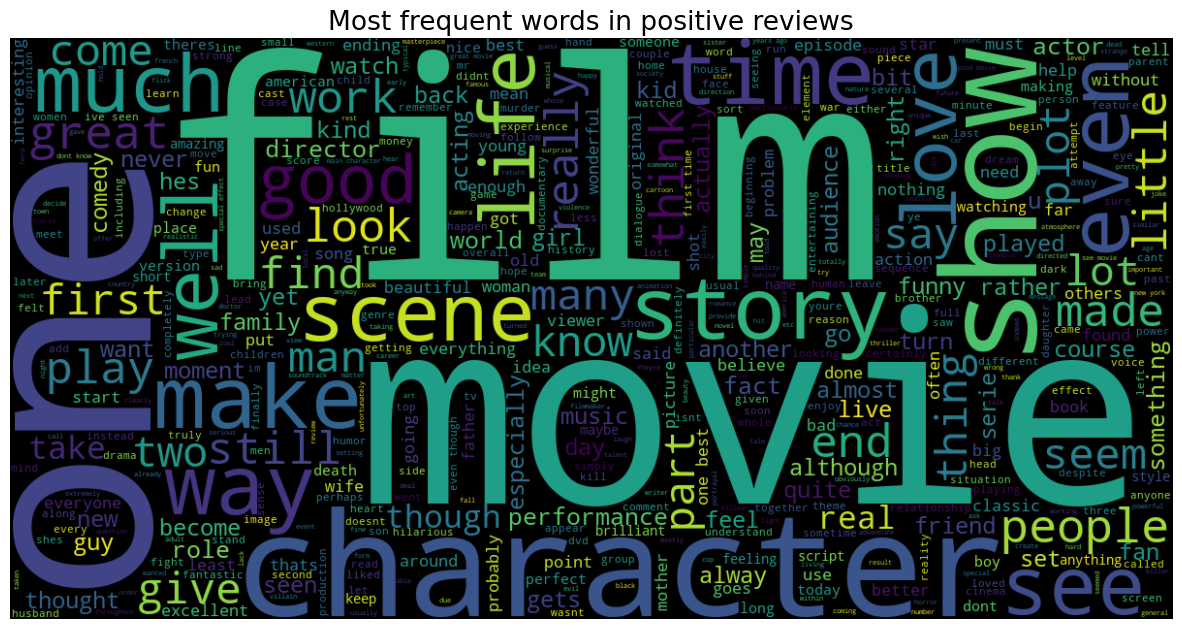


Fig. 11 Top Ten Average Ratings in IMDb Dataset.

**c. Must Frequent Words in Positive Reviews**

Fig. 12 shows the most common words found in texts about movies, with bigger words like film, movie, character, and scene appearing more frequently. Smaller words show terms that are employed less often, which aids in providing a brief overview of the primary subjects individuals address when films.

  
 Fig. 12 Must Frequent Words in Positive Reviews in IMDb Dataset.

**d.** **Must Frequent Words in Negative Reviews**

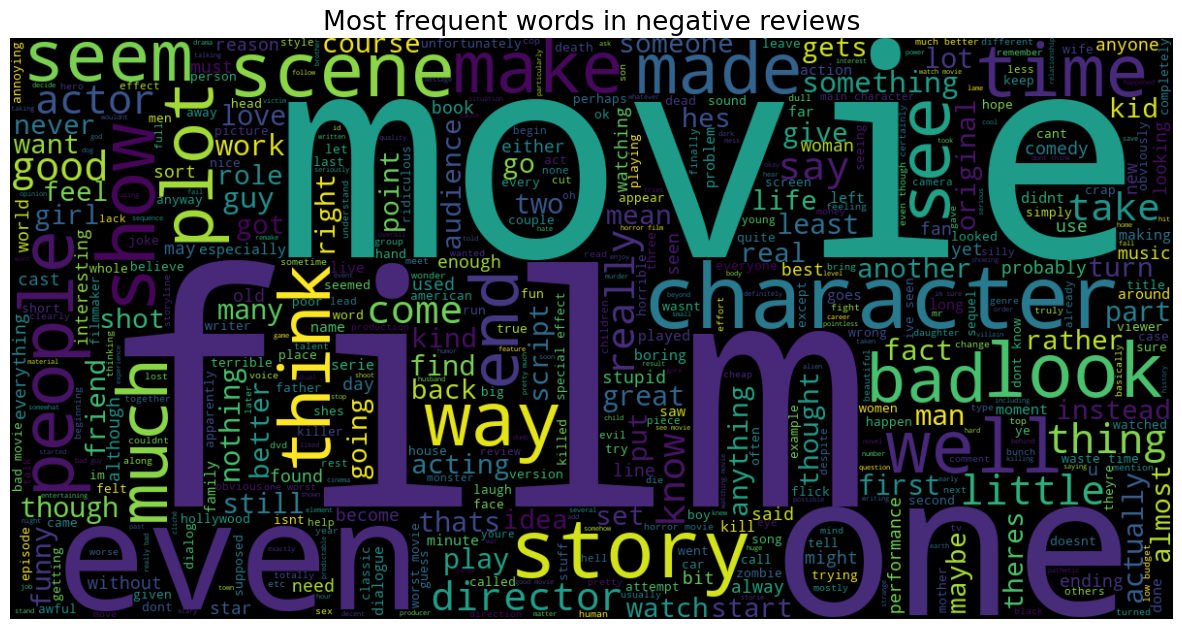


Fig. 13 Must Frequent Words in Negative Reviews in IMDb Dataset.

Fig. 13 illustrates the most frequent words individuals say when discussing films, with larger words such as film, movie, character, story, and scene being used more. Smaller terms such as director, role, and script are not mentioned as frequently, which informs us about what people are in reality discussing.

**e. Comparison of Two Models Accuracy**

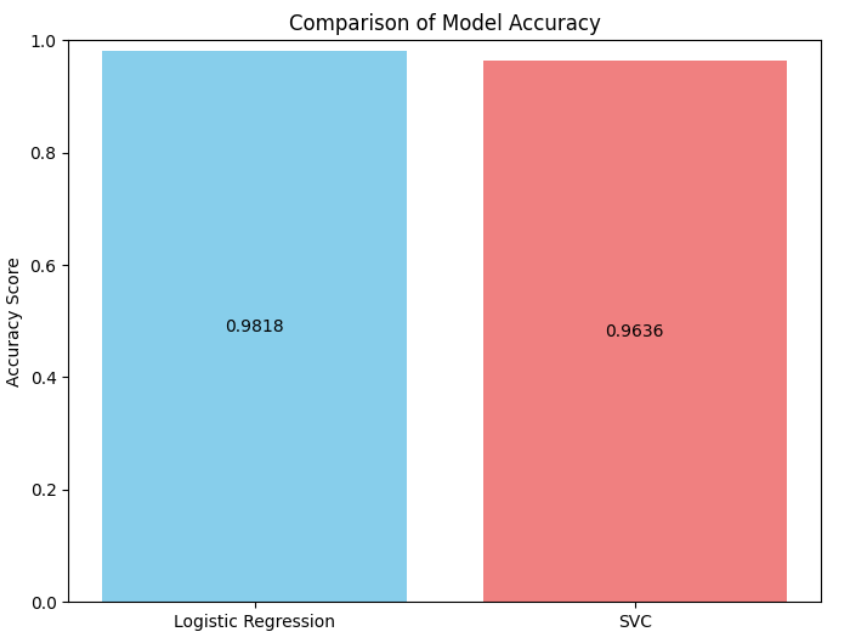


Fig. 14 Comparison of Two Models SVM and Logistic Regression

Fig. 14 illustrates graph Accuracy is a metric used to assess the extent to which a model or an algorithm is successful in terms of making accurate predictions of the cases and the formula is (TP + TN)/ (TP+TN+ FP+FN) where the TP and TN are correct predictions and FP and FN are errors. In this comparison, the accuracy of Logistic Regression was 98.18 percent, almost all being classified correctly, whereas the Support Vector Classifier (SVC) showed 96.36 percent which is also high except that it performed lower than the Logistic Regression. Accordingly, logical regression worked better than SVC in the correctness of prediction on this data set.

**7.3 NUMERICAL ANALYSIS**

A simulation of SVM and L R models in determining work result the movie's success based on rating and review was then evaluating using the measures symbolized by Acc, Prec, Reca and F1-Score.

Table 2. Models Performance Calculation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| Logistic Regression1 | 98.13% | 98% | 98% | 98% |
| Support Vector Machine2 | 96.36% | 96% | 96% | 96% |

**Performance Checking:**

* **Accuracy:** **(1)**

A model's accuracy is a performance metric that shows the proportion of accurate predictions it generates compared to all forecasts.

TP – True Positive.

TN – True Negative.

FP – False Positive.

FN – False Negative.

Measures the general correctness of the model; LR in our project attained 98.13% and SVM attained 96.36%, reflecting very high reliability.

* **Precision:**  **(2)**

The ratio of accurate positive predictions (True Positives / (True Positives + False Positives)) is known as precision.

TP – True Positive.

FP – False Positive.

Informs us about the number of movies identified as Hit actually being Hit; precision for both models was over 96%, reflecting low false positives.

* **Recall:**

**(3)**

The ratio of actual positive occurrences that the model properly predicted is called recall (True Positives / (True Positives + False Negatives)).

TP – True Positive.

FN – False Negative.

Shows how many positive Hit movies actually were correctly identified; LR slightly surpassed SVM, correctly capturing nearly all positive cases.

* **F1-Score:**

**(4)**

The F1 score is a machine learning assessment measure for classification models that computes the harmonic mean of accuracy and recall to provide a single number that strikes a balance between the two.

Balances Recall and Precision; at levels nearly 98% for LR and 96% for SVM, both models had very good consistency.

The comparison on these two methods: Logistic Regression and SVM reveal that both models performed quite well, yet, the first one appeared to perform a bit better than the last one. Logistic Regression attained an accuracy of 98.13% where precision, recall, and F1- score all equaled 98%. This was a reflection of having minimal misclassification and a balance of correct positive identification as well as some degree of the absence of false prediction. SVM also had an accuracy of 96.36 % with precisions, recalls, and F1-scores of 96% corresponding in strong performance though a bit lesser than Logistic Regression. Although the dependability rates of both models are quite good, Logistic Regression was found to be more consistent and accurate in the case of this dataset.

# CHAPTER 8

# FUTURE ENHANCEMENTS

**8.1 SUGGESTED IMPROVEMENTS AND ADDITIONAL FEATURES**

This work can be extended in the future using a larger and better-quality dataset comprising a wide variety of reviews (multilingual), coupled with financial indicators such as budget, revenue and eventually even cast and crew popularityThe actors and the actresses involved make a difference to a movie. Hyperparameter tuning may also be utilized to improve the methods efficiency as well as cross validation. and using ensemble methods or deep learning techniques like LSTMs/Transformers for more advanced sentiment analysis. Other elements like social media trend, genre-wise trends and seasonal release patterns could provide more enriched insights. For usability, we can build this into an interactive dashboard or, A web application that inset in viewers can insert their reviews of a movie and receive a real time prediction of whether it will be a hit or flop, along with explainability tools like Long Short-Term Memory so the users can see why a movie was classified a hit or flop. Finally, generalizing the problem for classification of non-binary multiple kinds of success (like a blockbuster movie, a common movie and a flopping) or even to prediction the box office amount that will be collected.

One of the key lessons of this project is that feature quality and diversity have a direct bearing on predictive model performance. While reviews and ratings already offered a solid basis for prediction, the results identified how much more precise and accurate the system could be using improved features. For instance, incorporating movie metadata like budget, cast and crew popularity, production house reputation, and timing of release would provide the model with more context on the drivers of success. In the same vein, including social media buzz, trailer engagement, and audience demographics may tap into real-time public interest that might not be indicated through reviews alone. More advanced natural language processing techniques, such as mood scores, emotion recognition, and keyword frequency, can offer additional indications even in the case of reviews. This would improve forecasting and build a more robust, more interpretable system closer to real-world decision-making in the movie business.

# CHAPTER 9

**CONCLUSION**

In future, this work can be enhanced by using a bigger and more diverse dataset, such as multilingual reviews, financial metrics like budget and revenue, and even cast and crew popularity because These are crucial in Fig.uring out a movie's success. The models' performance can be improved by hyperparameter tuning, cross-validation, and also by employing ensemble techniques or deep learning models like LSTMs and Transformers for sophisticated sentiment analysis. More features such as social media chatter, trends based on genres, and seasonal release patterns could generate more insights. From a user perspective, the project would be taken further to an interactive dashboard or web application in which users may enter reviews and ratings for real-time predictions, aided by explainability tools such as Local Interpretable Model-agnostic Explanations to identify why a film was rated as a hit or flop. Lastly, generalizing beyond binary classification to forecast several types of success (e.g., blockbuster, average, flop) or even predicting actual box office revenue would make the system more useful and relevant to the industry.

**REFRENCES**

[1] I. Steinke et al., Smart model to recognise emotions in movie reviews, IJACSA, Vol. 11, No. 5, 123-130, 2020.

[2] R Kumar and P Singh, Machine learning methods for sentiment analysis of IMDb movie reviews, International Journal of Computer Applications, Vol. 176, No. 2, pp. 25–30, 2021.

[3] R. Saputri and A. Baita, "Sentimen Review Film Avatar 2 pada platform IMDb menggunakan LSTM and GRU," Routers Journal Teknik Informatic Dan Terapan, Vol. 3, No. 1, 2025.

[4] A. M. Sarhan et al., "Machine learning, integration, and sentiment analysis of movies." In 2024, it will be published under the title, recommending the uses of systems, Journal Electrical Systems and Information Technology, Vol. 11, No. 1, 53, 2024.

[5] P. Atandoh, F. Zhang, M. A. Al-antari, D. Addo, and Y. H. Gu, "Scalable deep learning framework the sentiments analysis prediction of online film reviews," Heliyon, Vol. 10, No. 1, e30756, 2024..

[6] Steinke, I., Wier, J., Simon, L. and Seetan, R., 2022. Sentiment Analysis of Online films comments using Machine Learning. International Journal of Advanced Computer Science and Applications Vol. 13, No. 9, 2022.

[7] P. Baid, A. Gupta and N. Chaplot, Sentiment Analysis on Movie Reviews Used Machine Learning Techniques, International Journals of Computers Applications, Vol. 179, No. 7, 2017

[8] A. Timmaraju, V. Khanna, Sentiments Analysis on Movie Reviews Using Recursive and Recurrent Neural Network Architectures, Stanford University- Stanford, CA of USA, Project Report, Vol. 5, No 3, 2014.

[9] H. M. K. Kumar, B. S. Harish, and H. K. Darshan, Sentiment Analysis on the IMDb films feedbacks Using Hybrid Feature Extraction Method, International Journal of Management, Innovation & Economic Research, Vol. 5, No. 5, pp. 104-114 2018.

[10] Summarizing Online Movie Reviews: A Machine Learning Approach to Big Data Analytics, Scientific Programming, Vol. 4, No 13, Article ID 5812715, 2020

[11] Maureen J. Awan, Rubina A. Khan, Hamad Nobanee, Singh, et al. A Recommendation Engine to Predict Movie Ratings Using a Big Data Approach, Electronics 2021, 10, 1560, https://doi.org/10.3390/electronics10101560.

[12] Z. Mhowwala, A. R. Sulthana and S. D. Shetty, *Movie Ratings Prediction Use Ensemble Learning Methods,* Internationals Journal of Advanced Computer Science and Applications (IJACSA), Vol. 11, No. 8, 2020, pp. 383-388.

[13] A. Sharma and U. Ghose*, "Deep learning based bi-polar sentiment classification of movie reviews in Hindi,"* Journal of Statistics & Management Systems, Vol. 27, No. 1, pp. 59-86, 2024.

[14] D. Subedi, N. Lamichhane, and N. Subedi, Sentiment Analyzing of IMDb Movie comments Using SVM and Naive Bayes, Classifier, Journal of Engineering and Sciences, Vol. 4, No. 1, 57-64, May 2025.

[15] I. S. K. Wardhana, *“Analisa Internet Movie Database (Imdb) Menggunakan Algoritma Machine Learning Super Vector Machine,”* String (Satuan Tulisan Riset Dan Inovasi Teknologi), Vol. 9, No. 3, April 2025.

[16] M. Sanwal and C. Çalışkan, "A Hybrid Movie Recommender System and Rating Prediction Model," *International Journal of Information Technology and Applied Sciences*, Vol. 3, No. 3, pp. 161–168, Jul. 2021.

[17] S. Sahu, R. Kumar, M. S. Pathan, J. Shafi, Y. Kumar, and M. F. Ijaz,  
"Movie Popularity and Target Audience Prediction Using the Content-Based Recommender System," *IEEE Access*, Vol. 10, No. 3 pp. 42044–42058, Apr. 2022.

**PUBLICATION**

[1] Abdul Nasir and Hemanth Kumar “Review & Rating Based Film Sentiment Classification using Machine Learning Technique” International Journal of Multidisciplinary and Scientific Emerging Research (IJMSERH) Vol. 13, No. 3, pp. 1828-1835, Sept. 2025.