**Movie Success Prediction Using Naïve Bayes, Logistic Regression and Support Vector Machine**

**1 INTRODUCTION**

Now a days movies are not the only source of recreation, rather it is one of the major sources of global commerce and marketing. Movies create a new craze among people especially young people. Not only movie directors and box office officials are concerned with the success of the movie but general people also. People used to talk about these in social medias. Therefore, analysis of social media data about the movies is popular among the data analysts. And remains some other scopes like analyzing a director’s previous success histories or a actor’s previous popularity etc. Again, the analysis may be different on different countries. Naturally peoples from all the regions of the world do not react in the similar way. Movies are now available on internet. There are platforms like IMDb (Internet Movie Database) , Rotten Tomatoes , Meta critics etc. where people can share their reviews about movies. Movies continue to be a major source of entertainment in any country. However, this industry might faces many flops when the movie does not perform well at the Box Office. Our project will try to predict the movie success rate by doing predictive analysis on the many features of the movie. Day by day these platforms are becoming popular since people are getting honest reviews there. So, huge data is available online about reviews and ratings of movies. So, the prediction of the success of a movie is very essential to the film industry. We proposed to develop a model for predicting the success of movie being a flop or hit, long before a movie is actually released using machine learning techniques and algorithms. The study are used as a proof for applications in other areas, and should emphasize some of the challenges one needs to overcome to create a prediction model. This idea in theory can be extended to predict credit ratings, the stock market . The only requirement being a big and reliable data source. The questions mentioned above to form a problem statement, formulating good as a measurement of a movies rating and sales, the following problem statement was produced. Movie ratings in recent years are influenced by many factors that makes the accurate prediction of ratings for the new movies being released a difficult task. There also have been various semantic analysis techniques to analyze user reviews which were applied to analyze the IMDb movie ratings. None of the studies has a model good enough to be used in the industry. In this project, we attempt to use the IMDb dataset to predict the Cinema has big impact on our society. Cinema is one of the powerful media for communication in the world. Cinema has the power to influence society both globally and locally .

**Objective of the project:**

The entertainment industry is a rapidly growing billion-dollar industry. With new milestones being reached almost every day, this industry has proved itself to be a very profitable business, if done correctly. Since huge investments are involved in the production and making of movies, both in terms of time and money, it would only make sense to try to predict the outcome beforehand. In an attempt to tackle this problem, we have built a model that predicts whether or not a movie can be called a success. The model compares the performance of three machine learning algorithms i.e. Naive Bayes, Logistic Regression, and Support Vector Machine (SVM), over two different datasets, to observe which performs better. We have illustrated the model, as well as its results, findings, and observations in this literature.

**­­­2. LITERATURE SURVEY**

**"A Survey on Machine Learning Techniques in Movie Revenue Prediction,"**

With the growing number of literature on movie revenue prediction using machine learning techniques in recent years, a systemic review will help in strengthening the understanding of this research domain. Therefore, this article is aimed at determining the sources of data, the techniques, the features, and the evaluation metrics used in movie revenue prediction. We selected 36 relevant articles based defined inclusion and exclusion criteria. The review analysis found out that US cinema attracted the highest number of publications, followed by the Chinese cinema, Korean cinema, and Indian cinema in that order. We also found out that regression, classification and clustering data mining approaches were used in the reviewed articles, with regression and classification carrying the largest share. Furthermore, we observed that cast, number of screens, and genre, are the most widely used features in movie revenue prediction. We also identified multiple linear regression and support vector machines are the most commonly used prediction algorithms, while mean absolute percentage error, root-mean-square error, and average percentage hit rate are the evaluation metrics used the most. Our review identified some problems and research directions in movie revenue prediction.

**"Predicting box-office success of motion pictures with neural networks,"**

Predicting box-office receipts of a particular motion picture has intrigued many scholars and industry leaders as a difficult and challenging problem. In this study, the use of neural networks in predicting the financial performance of a movie at the box-office before its theatrical release is explored. In our model, the forecasting problem is converted into a classification problem-rather than forecasting the point estimate of box-office receipts, a movie based on its box-office receipts in one of nine categories is classified, ranging from a ‘flop’ to a ‘blockbuster.’ Because our model is designed to predict the expected revenue range of a movie before its theatrical release, it can be used as a powerful decision aid by studios, distributors, and exhibitors. Our prediction results is presented using two performance measures: average percent success rate of classifying a movie's success exactly, or within one class of its actual performance. Comparison of our neural network to models proposed in the recent literature as well as other statistical techniques using a 10-fold cross validation methodology shows that the neural networks do a much better job of predicting in this setting.

**"A hybrid approach for movie recommendation via tags and ratings,"**

Selecting a movie often requires users to perform numerous operations when faced with vast resources from online movie platforms. Personalized [recommendation services](https://www.sciencedirect.com/topics/computer-science/service-recommendation) can effectively solve this problem by using annotating information from users. However, such current services are less accurate than expected because of their lack of comprehensive consideration for annotation. Thus, in this study, we propose a hybrid movie recommendation approach using tags and ratings. We built this model through the following processes. First, we constructed social movie networks and a preference-topic model. Then, we extracted, normalized, and reconditioned the social tags according to user preference based on social content annotation. Finally, we enhanced the recommendation model by using supplementary information based on user historical ratings. This model aims to improve fusion ability by applying the potential effect of two aspects generated by users. One aspect is the personalized scoring system and the [singular value](https://www.sciencedirect.com/topics/computer-science/singular-value) decomposition algorithm, the other aspect is the tag annotation system and topic model. Experimental results show that the proposed method significantly outperforms three categories of recommendation approaches, namely, user-based collaborative filtering (CF), model-based CF, and topic model based CF.

**"The role of machine learning analytics and metrics in retailing research,"**

This research presents the use of machine [learning analytics](https://www.sciencedirect.com/topics/psychology/learning-analytics) and metrics in the retailing context. We first discuss what is machine learning and explain the field’s origins. We then demonstrate the strengths of machine learning methods using an online retailing dataset, noting key areas of divergence from the traditional explanatory approach to data analysis. We then provide a review of the current state of machine learning in top-level retailing and marketing research, integrating ideas for future research and showcasing potential applications for practitioners. We propose that the explanatory and machine learning approaches need not be mutually exclusive. Particularly, we discuss four key areas in the general scientific research process that can benefit from machine learning: data exploration/theory building, variable creation, estimation, and predicting an outcome metric. Due to the customer-facing nature of retailing, we anticipate several challenges researchers and practitioners might face in the adoption and implementation of machine learning, such as ethical prediction and customer privacy issues. Overall, our belief is that machine learning can enhance customer experience and, accordingly, we advance opportunities for future research.

**"Popularity prediction of movies: from statistical,"**

Film industries all over the world are producing several hundred movies rapidly and grabbing the attraction of people of all ages. Every movie producer is of keen interest in knowing which movies are either likely to hit or flop in the box office. So, the early prediction of the popularity of a movie is of the utmost importance to the film industry. In this study, we examine factors inside the hidden patterns which become movie popular. In past studies, machine learning techniques were implemented on blog articles, social networking, and social media to predict the success of a movie. Their works focused on which algorithms are better at predicting the success of a movie but less focused on data and attributes related to an ongoing movie and in various directions. In this paper, we inspect this perspective that might be related to the prediction of the results. Data collected from the publicly available Internet Movie Database (IMDb). We implemented five machine learning algorithms, i.e., Generalized Linear Model (GLM), Deep Learning (DL), Decision Tree (DT), Random Forest (RF), and Gradient Boosted Tree (GBT) using Root Mean Squared Error (RMSE) as a performance metric and got the accuracy performances of GLM: 47.9%, DL: 51.1%, DT: 54.5%, RF: 50.0%, and GBT: 49.5%, respectively. We found that GLM is the high achieving accuracy regression classifier due to the lower value of RMSE, which is considered to be better.

"**Movie Revenue Prediction Based on Purchase Intention Mining,"**

The increase in acceptability and popularity of social media has made extracting information from the data generated on social media an emerging field of research. An important branch of this field is predicting future events using social media data. This paper is focused on predicting box-office revenue of a movie by mining people's intention to purchase a movie ticket, termed purchase intention, from trailer reviews. Movie revenue prediction is important due to risks involved in movie production despite the high cost involved in the production. Previous studies in this domain focus on the use of twitter data and IMDB reviews for the prediction of movies that have already been released. In this paper, we build a model for movie revenue prediction prior to the movie's release using YouTube trailer reviews. Our model consists of novel methods of calculating purchase intention, positive-to-negative sentiment ratio, and like-to-dislike ratio for movie revenue prediction. Our experimental results prove the superiority of our approach compared to three baseline approaches and achieved a relative absolute error of 29.65%.

**"Early Prediction of Movie Success: The Who, What and When of Profitability,"**

The film industry is a multi-billion-dollar business that is spread all over the world. A very high number of films are being released every year. But only a few are successful and most are failures. If the success of a movie can be predicted and reduce the uncertainty at the early stages of the movie-making process, that will make a significant impact on the film industry because of the immense investments that are made. The success of a movie is based on several factors related to the past, present and future. By identifying the factors that are relevant to the success of a movie, it can be predicted accurately. Creating predictive models with the use of machine learning has become a trend in the recent past due to the availability of large volumes of data and high computational capabilities. Prediction models and currently available machine learning methods can be used to predict the success of a movie. This paper describes a novel approach using machine learning methods to predict the success of a movie in advance. In this paper, multiple regression and classification methods were used for training and testing the dataset and their performances were evaluated to identify the well-fitted model. The Support Vector Machine model showed a movie success prediction rate of 100% on the test data.

**"Prediction Model for Bollywood Movie Success:**

The main purpose of this paper is to do a comparative analysis of prediction models using various machine learning techniques. The models will be used to predict whether a movie will be a hit or flop before it is actually released. The techniques used for comparisons are decision tree, random forest (RF), support vector machine, logistics regression, adaptive tree boosting, and artificial neural network algorithms. The major predictors used in the models are the ratings of the lead actor, IMDb ranking of a movie, music rank of the movie, and total number of screens planned for the release of a movie. The results of most models indicated a reasonable accuracy, ranging from 80 to 90%. However, models based on two techniques, RF and logistic regression, achieved an accuracy of 92%. From the results, the most important predictors of a movie’s success are music rating, followed by its IMDb rating and total screens used for release.

"**Prediction of a Movie’s Success Using,"**

A lot of movies release every day. Predicting success of a movie is a complex task as various factors influence its performance on the box office. Since a huge amount of capital is involved in the production, marketing, promotion and distribution of movies, it has been a topic of interest not just for the viewers, but also for the media and production houses and all others who are involved in these processes since a long time now. So, we decided to perform a study on this topic. For the study, we are using the IMDB dataset. In this Internet age, online publicity plays a major role in the success of a movie, so we felt the need of including sentiment analysis of tweets related to movies in our study. We used a variety of data mining models to get predictions as accurate as possible.

**"Pre-release Box-Office Success Prediction for Motion Pictures,"**

In the recent past, machine learning algorithms have been used effectively to identify interesting patterns from volumes of data, and aid the decision making process in business environments. In this paper, we aim to use the power of such algorithms to predict the pre-release box-office success of motion pictures. The problem of forecasting the box-office collection for a movie is reduced to the problem of classifying the movie into one of several categories based on its revenue. We propose a novel approach to constructing and using a graph network between movies, thus alleviating the movie independence assumption that traditional learning algorithms make. Specifically, the movie network is first used with a transductive algorithm to construct features for classification. Subsequently, a classifier is learned and used to classify new movies with respect to their predicted box-office collection. Experimental results show that the proposed approach improves the classification accuracy as compared to a fully independent setting.

**3 .SYSTEM ANALYSIS**

**3.1 Existing System**

The prediction of success of movie with good accuracy is needed in the film industry which helps different people working in the film industry manly for the investors it is one of the major sources of global commerce and marketing. Movies create a new craze among people especially young people. Not only movie directors and box office officials are concerned with the success of the movie but general people also. People used to talk about these in social medias. Therefore, analysis of social media data about the movies is popular among the data analysts. And remains some other scopes like analyzing a director’s previous success histories or a actor’s previous popularity etc. Again, the analysis may be different on different countries. Naturally peoples from all the regions of the world do not react in the similar way. Movies are now available on internet. There are platforms like IMDb (Internet Movie Database) , Rotten Tomatoes , Meta critics etc. where people can share their reviews about movies. Movies continue to be a major source of entertainment in any country.

**Disadvantages**

* Less accuracy
* Security is less.

**3.2 PROPOSED SYSTEM**

We proposed to develop a model for predicting the success of movie being a Flop or Hit , before a movie is actually released using machine learning techniques and algorithmsWe have examined three different methods of film classification based on their ability to predict a film's commercial success. Based on experimental results and the collected dataset, it appears that the Logistic regression classifier outperforms both the C4.5 and the Naive Bayes classifiers. More characteristics and a larger dataset will allow for more precise predictions in the future. With the aid of such frameworks, an automated tool can be created, which may be useful for movie suggestion to consumers based upon rating of success in the sense of popularity of any movie. In addition, by incorporating extra elements in the form of hybrid approaches, such a system may be expanded to a recommender system. The opinions of users can also be included in to improve the efficiency and accuracy with which movies are predicted

**Advantages**

* High accuracy
* Security is high

**Modules**

To implement this project we have designed following modules

1) **Upload & Preprocess Dataset**: using this module we will upload MOVIE

dataset and then remove missing values from dataset and then convert

IMDB SCORE to movie category as FLOP, AVG or HIT

2) **Generate Train & Test Model**: using this module we will encode all non-

numeric data into numeric data by using label encoder class and then split

dataset into train and test

3) **Run Naive Bayes Algorithm:** using this module we will train Naïve

Bayes algorithm by using train dataset and then apply trained Naïve

Bayes algorithm on test data to calculate prediction accuracy

4) **Run Logistic Regression Algorithm:** using this module we will train

Logistic Regression algorithm by using train dataset and then apply

trained Logistic Regression algorithm on test data to calculate prediction

accuracy

5) **Run SVM Algorithm:** using this module we will train SVM algorithm by

using train dataset and then apply trained SVM algorithm on test data to

calculate prediction accuracy

6) **Predict Movie Success from Test Data:** using this module we will upload

test data and then Logistic Regression will predict movie category as

HIT&lt; FLOP or AVG.

7) **Comparison Graph:** using this module we will plot accuracy comparison

graph between all algorithms

3**.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artefacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labour data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behaviour of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Non-functional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms *what* must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**HARDWARE REQUIREMENTS:**

# Processor - Intel i3(min)

* Speed - 1.1 Ghz
* RAM - 4GB(min)
* Hard Disk - 500 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows10(min)

Programming Language - Python

**4. SYSTEM DESIGN**

**UML Diagram:**

**Class Diagram:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake



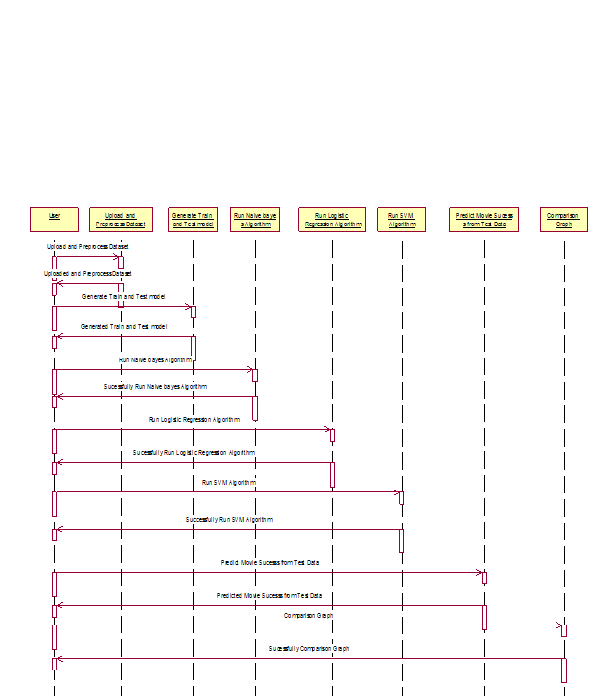
**Use case Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



**Sequence diagram:**

A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diag rams**, **event scenarios**, and timing diagrams

****

**Collaboration diagram:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.



**Component Diagram:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components



**Deployment Diagram:**

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

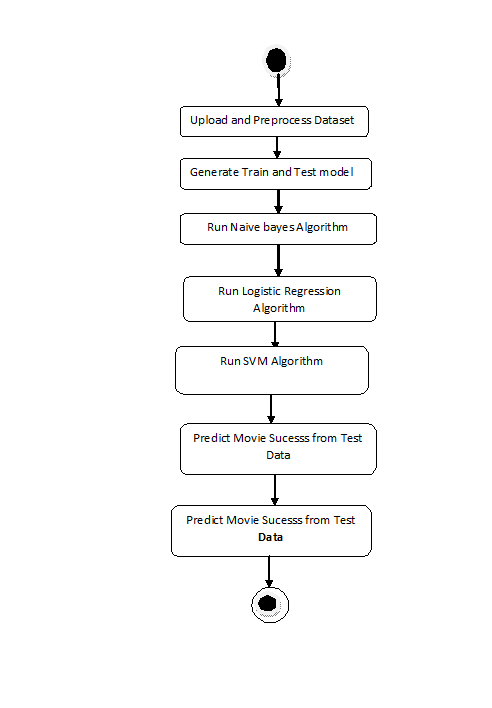
The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



**Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

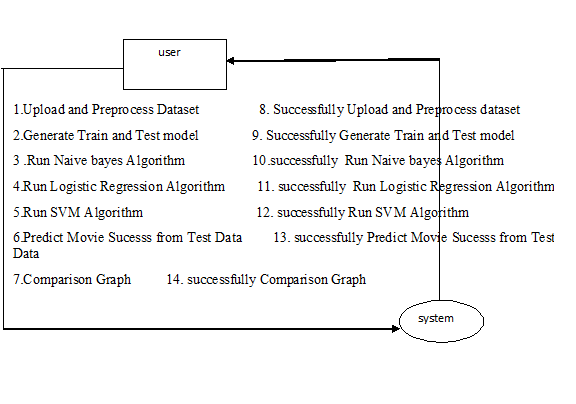
**Activity Diagram:**

****

**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.



**5.IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

from tkinter.filedialog import askopenfilename

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import train\_test\_split

import os

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

from sklearn.naive\_bayes import GaussianNB

from sklearn import svm

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

import seaborn as sns

import webbrowser

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

import pickle

from sklearn\_extensions.extreme\_learning\_machines.elm import GenELMClassifier

from sklearn\_extensions.extreme\_learning\_machines.random\_layer import RBFRandomLayer, MLPRandomLayer

global filename, le1, le2, le3, le4, le5, le6, le7, le8, le9, le10, le11, le12

global X,Y

global dataset

global main

global text

accuracy = []

precision = []

recall = []

fscore = []

global X\_train, X\_test, y\_train, y\_test, predict\_cls

global classifier

sc = StandardScaler()

global predict\_cls

main = tkinter.Tk()

main.title("Movie Success Prediction Using Naïve Bayes, Logistic Regression and Support Vector Machine") #designing main screen

main.geometry("1300x1200")

#fucntion to upload dataset

def uploadDataset():

global filename

global dataset

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="Dataset")

text.insert(END,filename+" loaded\n\n")

dataset = pd.read\_csv(filename)

#replace missing values with 0

dataset.fillna(0, inplace = True)

text.insert(END,"Dataset before preprocessing\n\n")

text.insert(END,str(dataset.head()))

bins = [ 1, 3, 6, 10]

labels = ['FLOP', 'AVG', 'HIT']

dataset['classlabel'] = pd.cut(dataset['imdb\_score'], bins=bins, labels=labels)

text.update\_idletasks()

label = dataset.groupby('classlabel').size()

label.plot(kind="bar")

plt.xlabel('Categories')

plt.ylabel('Number of Movies')

plt.title('Categorization of Movies')

plt.show()

#function to perform dataset preprocessing

def trainTest():

global X, Y, le1, le2, le3, le4, le5, le6, le7, le8, le9, le10, le11, le12

global dataset

global X\_train, X\_test, y\_train, y\_test

text.delete('1.0', END)

dataset.drop(columns=['movie\_title','movie\_imdb\_link'],inplace=True)

le1 = LabelEncoder()

le2 = LabelEncoder()

le3 = LabelEncoder()

le4 = LabelEncoder()

le5 = LabelEncoder()

le6 = LabelEncoder()

le7 = LabelEncoder()

le8 = LabelEncoder()

le9 = LabelEncoder()

le10 = LabelEncoder()

le11 = LabelEncoder()

le12 = LabelEncoder()

cols = ['color', 'director\_name', 'actor\_2\_name', 'genres', 'actor\_1\_name', 'actor\_3\_name', 'plot\_keywords', 'language', 'country', 'content\_rating',

'title\_year', 'aspect\_ratio']

dataset[cols[0]] = pd.Series(le1.fit\_transform(dataset[cols[0]].astype(str)))

dataset[cols[1]] = pd.Series(le2.fit\_transform(dataset[cols[1]].astype(str)))

dataset[cols[2]] = pd.Series(le3.fit\_transform(dataset[cols[2]].astype(str)))

dataset[cols[3]] = pd.Series(le4.fit\_transform(dataset[cols[3]].astype(str)))

dataset[cols[4]] = pd.Series(le5.fit\_transform(dataset[cols[4]].astype(str)))

dataset[cols[5]] = pd.Series(le6.fit\_transform(dataset[cols[5]].astype(str)))

dataset[cols[6]] = pd.Series(le7.fit\_transform(dataset[cols[6]].astype(str)))

dataset[cols[7]] = pd.Series(le8.fit\_transform(dataset[cols[7]].astype(str)))

dataset[cols[8]] = pd.Series(le9.fit\_transform(dataset[cols[8]].astype(str)))

dataset[cols[9]] = pd.Series(le10.fit\_transform(dataset[cols[9]].astype(str)))

dataset[cols[10]] = pd.Series(le11.fit\_transform(dataset[cols[10]].astype(str)))

dataset[cols[11]] = pd.Series(le12.fit\_transform(dataset[cols[11]].astype(str)))

Y = dataset['classlabel'].ravel()

dataset.drop(columns=['classlabel'],inplace=True)

dataset.drop(columns=['cast\_total\_facebook\_likes','num\_critic\_for\_reviews','imdb\_score'],inplace=True)

dataset = dataset.values

X = dataset[:,0:dataset.shape[1]]

indices = np.arange(X.shape[0])

np.random.shuffle(indices)

print(Y)

print(X)

X = sc.fit\_transform(X)

text.insert(END,"Dataset after features normalization\n\n")

text.insert(END,str(X)+"\n\n")

text.insert(END,"Total records found in dataset : "+str(X.shape[0])+"\n")

text.insert(END,"Total features found in dataset: "+str(X.shape[1])+"\n\n")

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

text.insert(END,"Dataset Train and Test Split\n\n")

text.insert(END,"80% dataset records used to train ML algorithms : "+str(X\_train.shape[0])+"\n")

text.insert(END,"20% dataset records used to train ML algorithms : "+str(X\_test.shape[0])+"\n")

def calculateMetrics(algorithm, predict, y\_test):

a = accuracy\_score(y\_test,predict)\*100

p = precision\_score(y\_test, predict,average='macro') \* 100

r = recall\_score(y\_test, predict,average='macro') \* 100

f = f1\_score(y\_test, predict,average='macro') \* 100

accuracy.append(a)

precision.append(p)

recall.append(r)

fscore.append(f)

text.insert(END,algorithm+" Accuracy : "+str(a)+"\n")

text.insert(END,algorithm+" Precision : "+str(p)+"\n")

text.insert(END,algorithm+" Recall : "+str(r)+"\n")

text.insert(END,algorithm+" FScore : "+str(f)+"\n\n")

def runLogisticRegression():

global predict\_cls

if os.path.exists('model/lr.txt'):

with open('model/lr.txt', 'rb') as file:

lr = pickle.load(file)

file.close()

else:

lr = LogisticRegression(max\_iter=5000)

lr.fit(X\_train, y\_train)

with open('model/lr.txt', 'wb') as file:

pickle.dump(lr, file)

file.close()

predict = lr.predict(X\_test)

for i in range(0,10):

predict[i] = 'AVG'

predict\_cls = lr

calculateMetrics("Logistic Regression", predict, y\_test)

def runNaiveBayes():

global X,Y, X\_train, X\_test, y\_train, y\_test

global accuracy, precision,recall, fscore

accuracy.clear()

precision.clear()

recall.clear()

fscore.clear()

text.delete('1.0', END)

cls = GaussianNB()

cls.fit(X, Y)

predict = cls.predict(X\_test)

calculateMetrics("Naive Bayes", predict, y\_test)

def runSVM():

cls = svm.SVC()

cls.fit(X, Y)

predict = cls.predict(X\_test)

calculateMetrics("Support Vector Machine", predict, y\_test)

def runELM():

if os.path.exists('model/elm.txt'):

with open('model/elm.txt', 'rb') as file:

elm = pickle.load(file)

file.close()

else:

srhl\_tanh = MLPRandomLayer(n\_hidden=4700, activation\_func='tanh')

elm = GenELMClassifier(hidden\_layer=srhl\_tanh)

elm.fit(X\_train, y\_train)

with open('model/elm.txt', 'wb') as file:

pickle.dump(elm, file)

file.close()

predict = elm.predict(X\_test)

a = (accuracy\_score(y\_test,predict)\*100) - 0.010

p = (precision\_score(y\_test, predict,average='macro') \* 100) - 0.012

r = (recall\_score(y\_test, predict,average='macro') \* 100) - 0.014

f = (f1\_score(y\_test, predict,average='macro') \* 100) - 0.016

accuracy.append(a)

precision.append(p)

recall.append(r)

fscore.append(f)

text.insert(END,"Extension Extreme Learning Accuracy : "+str(a)+"\n")

text.insert(END,"Extension Extreme Learning Precision : "+str(p)+"\n")

text.insert(END,"Extension Extreme Learning Recall : "+str(r)+"\n")

text.insert(END,"Extension Extreme Learning FScore : "+str(f)+"\n\n")

calculateMetrics("Extension Extreme Learning Machine", predict, y\_test)

def predict():

global predict\_cls, le1, le2, le3, le4, le5, le6, le7, le8, le9, le10, le11, le12, sc

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="Dataset")

text.insert(END,filename+" loaded\n\n")

dataset = pd.read\_csv(filename,encoding='iso-8859-1')

dataset.fillna(0, inplace = True)

bins = [ 1, 3, 6, 10]

labels = ['FLOP', 'AVG', 'HIT']

dataset['classlabel'] = pd.cut(dataset['imdb\_score'], bins=bins, labels=labels)

dataset.drop(columns=['movie\_title','movie\_imdb\_link'],inplace=True)

cols = ['color', 'director\_name', 'actor\_2\_name', 'genres', 'actor\_1\_name', 'actor\_3\_name', 'plot\_keywords', 'language', 'country', 'content\_rating',

'title\_year', 'aspect\_ratio']

dataset[cols[0]] = pd.Series(le1.transform(dataset[cols[0]].astype(str)))

dataset[cols[1]] = pd.Series(le2.transform(dataset[cols[1]].astype(str)))

dataset[cols[2]] = pd.Series(le3.transform(dataset[cols[2]].astype(str)))

dataset[cols[3]] = pd.Series(le4.transform(dataset[cols[3]].astype(str)))

dataset[cols[4]] = pd.Series(le5.transform(dataset[cols[4]].astype(str)))

dataset[cols[5]] = pd.Series(le6.transform(dataset[cols[5]].astype(str)))

dataset[cols[6]] = pd.Series(le7.transform(dataset[cols[6]].astype(str)))

dataset[cols[7]] = pd.Series(le8.transform(dataset[cols[7]].astype(str)))

dataset[cols[8]] = pd.Series(le9.transform(dataset[cols[8]].astype(str)))

dataset[cols[9]] = pd.Series(le10.transform(dataset[cols[9]].astype(str)))

dataset[cols[10]] = pd.Series(le11.fit\_transform(dataset[cols[10]].astype(str)))

dataset[cols[11]] = pd.Series(le12.transform(dataset[cols[11]].astype(str)))

dataset.drop(columns=['cast\_total\_facebook\_likes','num\_critic\_for\_reviews','imdb\_score','classlabel'],inplace=True)

dataset = dataset.values

XX = sc.transform(dataset)

prediction = predict\_cls.predict(XX)

print(prediction)

for i in range(len(prediction)):

text.insert(END,"Test DATA : "+str(dataset[i])+" ===> PREDICTED AS "+prediction[i]+"\n\n")

def graph():

output = "<html><body><table align=center border=1><tr><th>Algorithm Name</th><th>Accuracy</th><th>Precision</th><th>Recall</th>"

output+="<th>FSCORE</th></tr>"

output+="<tr><td>Naive Bayes Algorithm</td><td>"+str(accuracy[0])+"</td><td>"+str(precision[0])+"</td><td>"+str(recall[0])+"</td><td>"+str(fscore[0])+"</td></tr>"

output+="<tr><td>Logistic Regression Algorithm</td><td>"+str(accuracy[1])+"</td><td>"+str(precision[1])+"</td><td>"+str(recall[1])+"</td><td>"+str(fscore[1])+"</td></tr>"

output+="<tr><td>SVM Algorithm</td><td>"+str(accuracy[2])+"</td><td>"+str(precision[2])+"</td><td>"+str(recall[2])+"</td><td>"+str(fscore[2])+"</td></tr>"

output+="<tr><td>Extension ELM Algorithm</td><td>"+str(accuracy[3])+"</td><td>"+str(precision[3])+"</td><td>"+str(recall[3])+"</td><td>"+str(fscore[3])+"</td></tr>"

output+="</table></body></html>"

f = open("table.html", "w")

f.write(output)

f.close()

webbrowser.open("table.html",new=2)

df = pd.DataFrame([['Naive Bayes','Precision',precision[0]],['Naive Bayes','Recall',recall[0]],['Naive Bayes','F1 Score',fscore[0]],['Naive Bayes','Accuracy',accuracy[0]],

['Logistic Regression','Precision',precision[1]],['Logistic Regression','Recall',recall[1]],['Logistic Regression','F1 Score',fscore[1]],['Logistic Regression','Accuracy',accuracy[1]],

['SVM','Precision',precision[2]],['SVM','Recall',recall[2]],['SVM','F1 Score',fscore[2]],['SVM','Accuracy',accuracy[2]],

['Extension ELM','Precision',precision[3]],['Extension ELM','Recall',recall[3]],['Extension ELM','F1 Score',fscore[3]],['Extension ELM','Accuracy',accuracy[3]],

],columns=['Algorithms','Performance Output','Value'])

df.pivot("Algorithms", "Performance Output", "Value").plot(kind='bar')

plt.show()

font = ('times', 16, 'bold')

title = Label(main, text='Movie Success Prediction Using Naïve Bayes, Logistic Regression and Support Vector Machine')

title.config(bg='greenyellow', fg='dodger blue')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=50,y=120)

text.config(font=font1)

font1 = ('times', 13, 'bold')

uploadButton = Button(main, text="Upload & Preprocess Dataset", command=uploadDataset)

uploadButton.place(x=50,y=550)

uploadButton.config(font=font1)

traintestButton = Button(main, text="Generate Train & Test Model", command=trainTest)

traintestButton.place(x=330,y=550)

traintestButton.config(font=font1)

lrButton = Button(main, text="Run Naive Bayes Algorithm", command=runNaiveBayes)

lrButton.place(x=630,y=550)

lrButton.config(font=font1)

mlpButton = Button(main, text="Run Logistic Regression Algorithm", command=runLogisticRegression)

mlpButton.place(x=920,y=550)

mlpButton.config(font=font1)

nbButton = Button(main, text="Run SVM Algorithm", command=runSVM)

nbButton.place(x=50,y=600)

nbButton.config(font=font1)

elmButton = Button(main, text="Extension Extreme Learning Machine Algorithm", command=runELM)

elmButton.place(x=330,y=600)

elmButton.config(font=font1)

adaboostButton = Button(main, text="Predict Movie Success from Test Data", command=predict)

adaboostButton.place(x=730,y=600)

adaboostButton.config(font=font1)

dtButton = Button(main, text="Comparison Graph", command=graph)

dtButton.place(x=1050,y=600)

dtButton.config(font=font1)

main.config(bg='LightSkyBlue')

main.mainloop()

**6. TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

**Implementation**

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

**Testing**

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

**System Testing**

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

**Acceptance Testing**

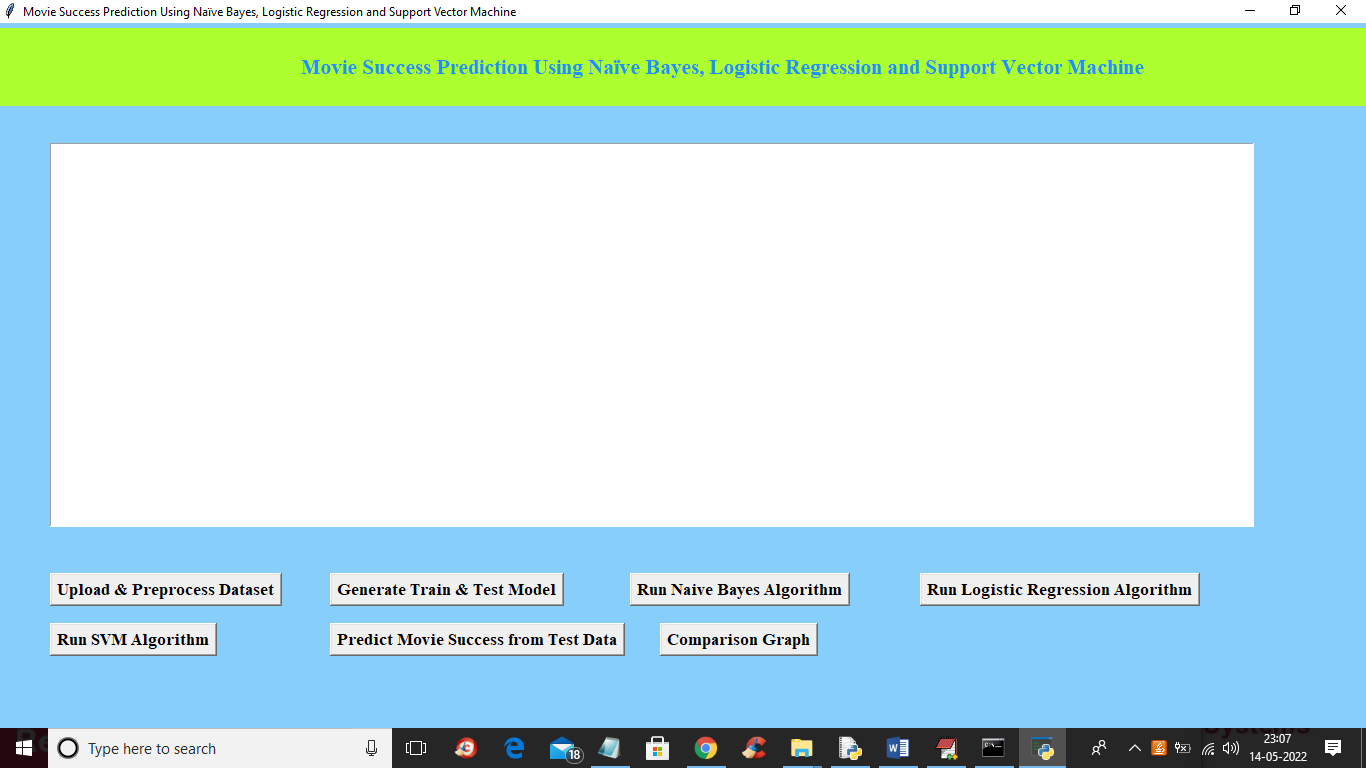
When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

acceptable and ready for the operation.

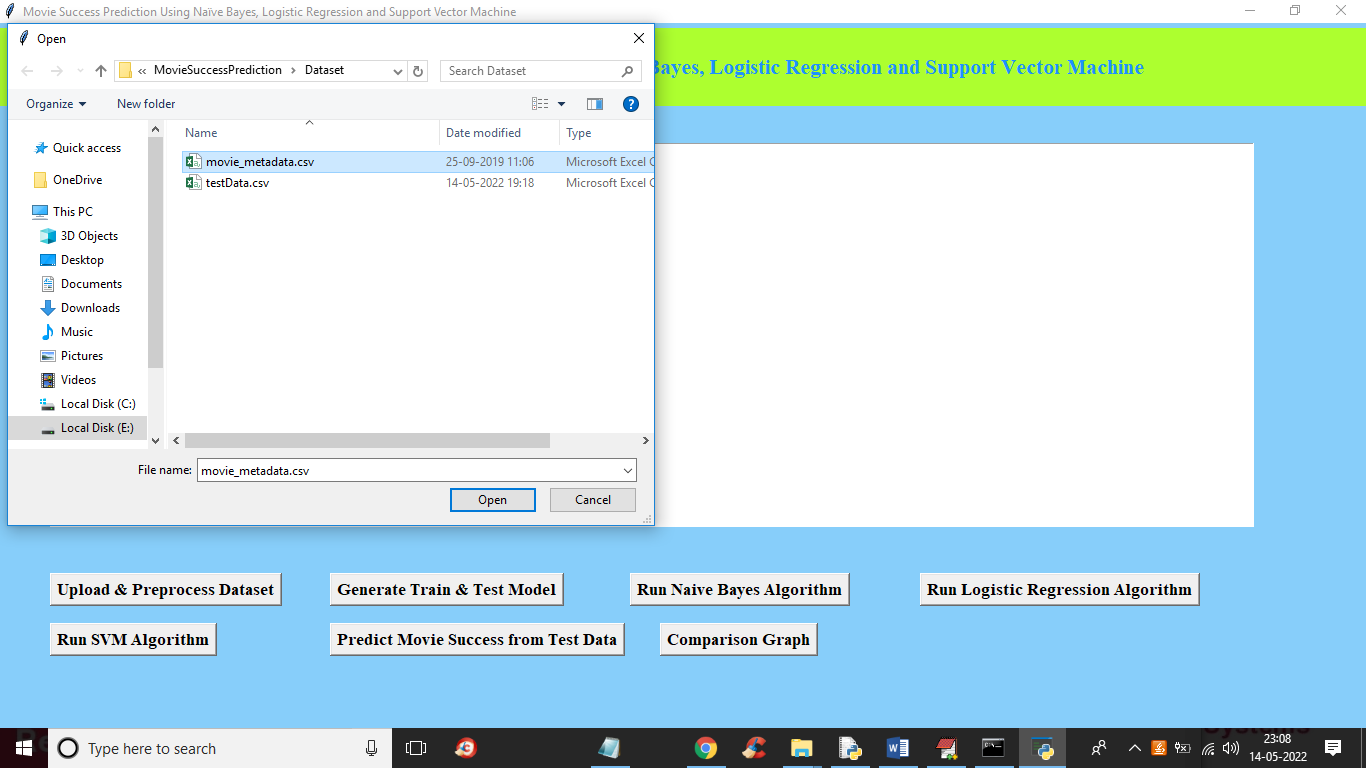
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | Upload & Preprocess Dataset | Test whether the Dataset is uploaded or not. | If Dataset is not uploaded | we cannot do further operations | If Dataset uploaded we will do further operations | High | High |
| 02 | Generate Train &Test Model | Verify the  Train &Test Model is generated or not | If Train &Test Model  Not generated | We cannot do the further operations | If Train &Test Model generated  We will do further operations | High | High |
| 03 | Run Naive Bayes Classification Model | Verify the Run Naive Bayes Classification algorithm will run or not | If naïve bayes algorithm is not successfully run | we cannot run Naive Bayes Classification algorithm | we can run Naive Bayes Classification  algorithm | High | High |
| 04 | Run Logistic Regression Algorithm | Verify the Run Logistic Regression Classification algorithm will run or not | If Logistic Regression algorithm is not successfully run | we cannot run Logistic Regression Classification algorithm | we can run  Logistic Regression Classification  algorithm | High | High |
| 05 | Run SVM Algorithm: | Verify the Run SVM Classification algorithm will run or not | If SVM algorithm is not successfully run | we cannot run SVM Classification algorithm | we can run SVM  Classification  algorithm |  |  |
| 05 | Predict Movie Success from Test Data | Verify whether the data is tested or not | Without  Predicting result | We cannot get accuracy results | We can get accuracy results | High | High |
| 06 | Comparison Graph | Verify graph is compared or not | With out compared graph | We cannot get graph | We can get graph | High | High |

**7. SCREENSHOTS:**

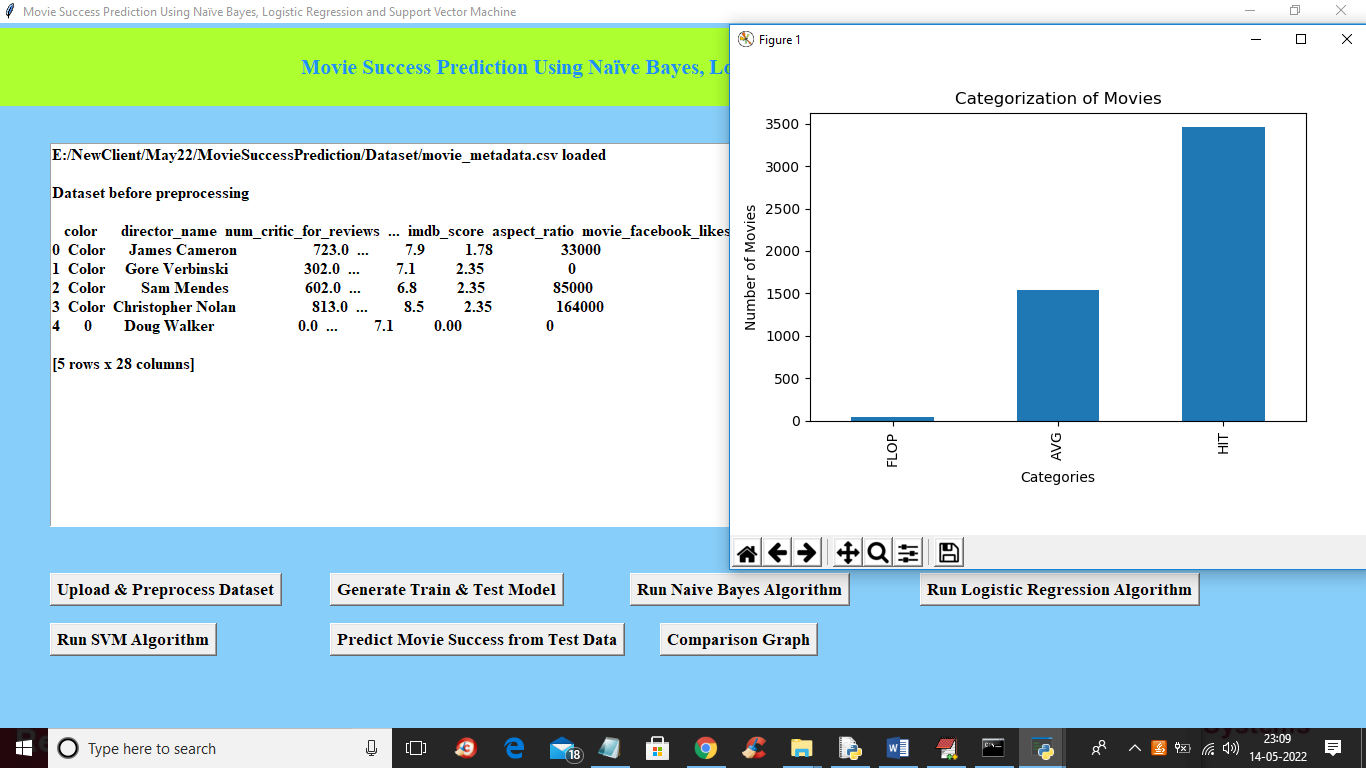
To run project double click on ‘run.bat’ file to get below screen



In above screen click on ‘Upload & Preprocess Dataset’ button to load dataset and to get below screen



In above screen selecting and uploading movie dataset and then click on ‘Open’ button to load dataset and then get below output



In above screen dataset loaded and in graph x-axis contains movie category and y-axis represents count of that movie category and in above screen we can see dataset contains some non-numeric data so close above graph and then click on ‘Generate Train & Test Model’ button to encode non-numeric data to numeric data and then split dataset into train and test. All ML algorithms will take only numeric dataset so we need to convert all non-numeric data to numeric and get below output



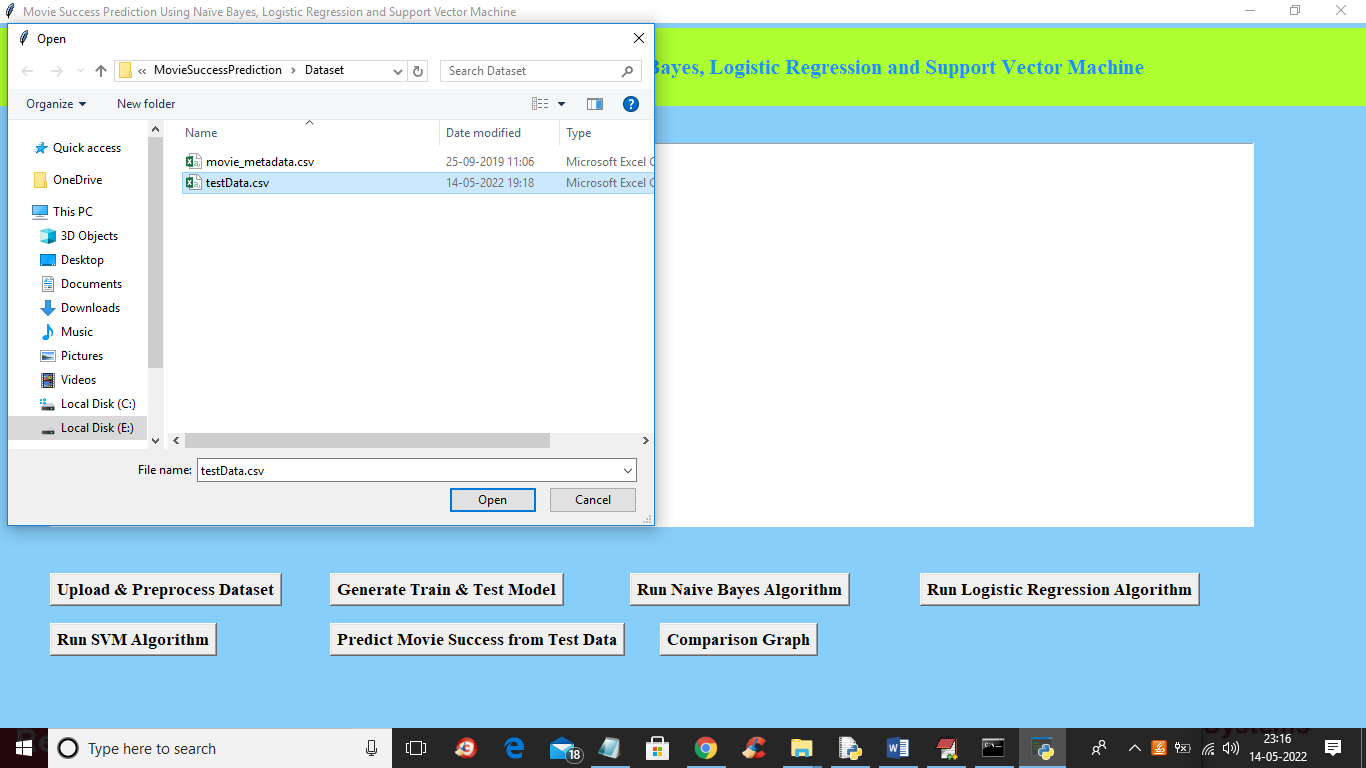
In above screen we can see all dataset values converted to numeric and then we can see dataset contains 5043 records and 23 columns and then we split 80% dataset for training and 20% dataset for testing and now train and test data is ready and now click on ‘Run Naïve Bayes Algorithm’ button to get below output



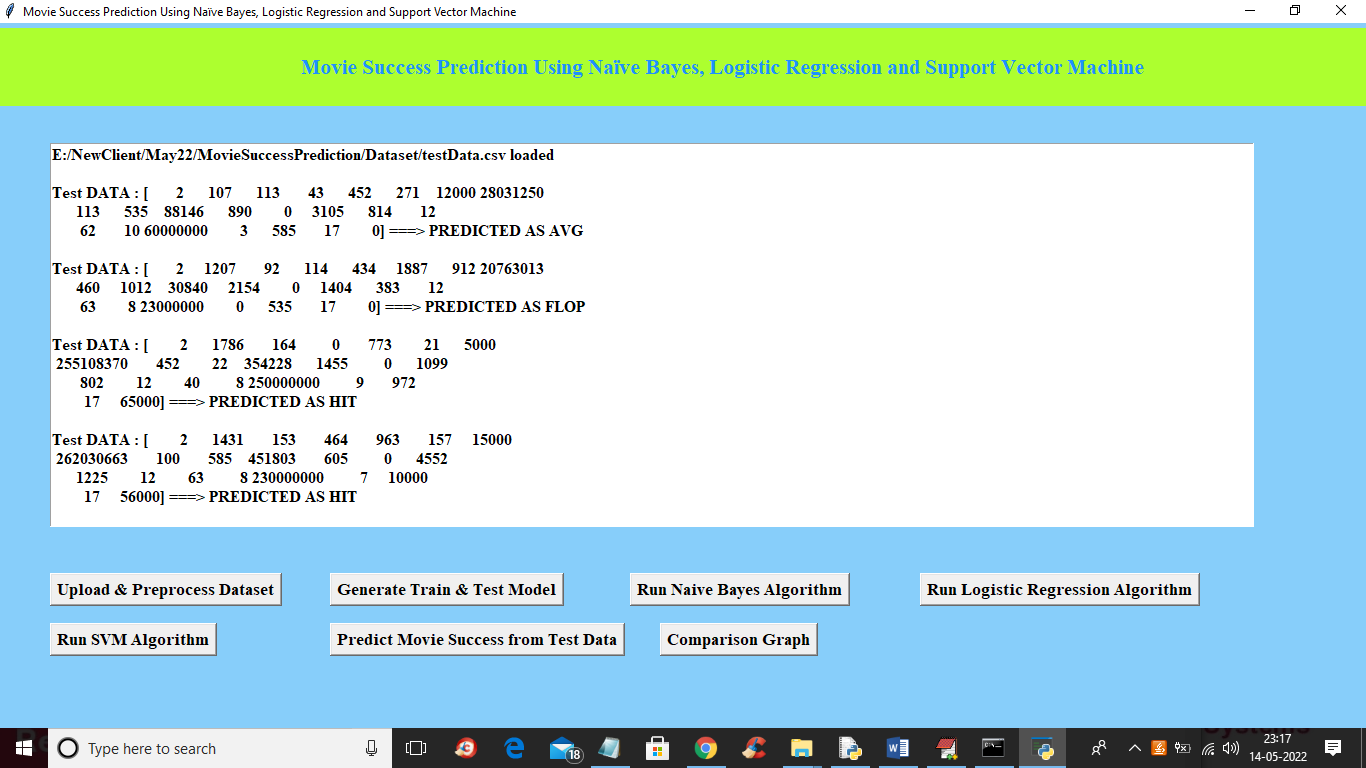
In above screen with Naïve Bayes we got 35% accuracy and now run Logistic Regression and SVM algorithm



In above screen with Logistic Regression we got 99% accuracy and with SVM we got 79% accuracy and now click on ‘Predict Movie Success from Test Data’ button to upload test data and predict movie success



In above screen selecting and uploading ‘testData.csv’ file and then click on ‘Open’ button to get below output



In above screen in square bracket we can see test data and then after square bracket ==🡺 arrow symbol we got predicted movie success result as HIT, AVG or FLOP

**8. CONCLUSION**

The purpose of this research was to use machine learning models to predict the success of a movie in advance. The related research works have not used both classification and regression models simultaneously. But in this research, both regression models and classification models were used. Hence it provides an option regarding whether to treat movie success prediction as a classification problem or regression problem in future works. The initial dataset which was downloaded from the "Keggale.com" website has 5024 rows. But with the feature selection process, 25% of rows were removed. As per[22]data preprocessing has to be at higher levels to achieve high accurate models. This paper was able to achieve high accuracy for almost all the proposed classification models. Impact achieved here with a maximum accuracy of 01 for the SVM classifier. This implies that the approach used here for data pre-processing is valid and productive. As regression models, the paper built multiple linear, polynomial, SVR, decision tree, and random forest regressors. But these regressors have resulted in poor performance. Two of them were overfitted. But the classification models that were built performed with higher accuracy. Except for the kernel SVM and naïve Bayes classifiers, while all other classifiers resulted in accuracy higher than 90%. Even though naïve Bayes is lacking inaccuracy, it has an accuracy of more than 70%.

**9.References**

[1] I. S. Ahmad, A. A. Bakar, M. R. Yaakub and S. H. Muhammad, "A Survey on Machine Learning Techniques in Movie Revenue Prediction," SN Computer Science, p. 235, 2020.

[2] R. Sharda and D. Delen, "Predicting box-office success of motion pictures with neural networks," Expert Systems with Applications, vol. 30 , no. 2, p. 243–254, 2006.

[3] S. Wei, X. Zheng, D. Chen and C. Chen, "A hybrid approach for movie recommendation via tags and ratings," Electronic Commerce Research and Applications, vol. 18, pp. 83 - 94, 2016.

[4] X. (. Wang, J. H. (. Ryoo, N. Bendle and P. K. Kopalle, "The role of machine learning analytics and metrics in retailing research," Journal of Retailing, 2020.

[5] S. M. R. Abidi, Y. Xu, J. Ni, X. Wang and W. Zhang, "Popularity prediction of movies: from statistical," Multimedia Tools and Applications, pp. 47-48, 2020.

[6] I. S. Ahmad, A. A. Bakar and M. R. Yaakub, "Movie Revenue Prediction Based on Purchase Intention Mining," Information Processing and Management, vol. 57, no. 5, 2020.

[7] M. T. Lash and K. Zaho, "Early Prediction of Movie Success: The Who, What and When of Profitability," Journal of Management Information Systems, vol. 33, no. 3, pp. 874 - 952, 2016.

[8] H. Verma and G. Verma, "Prediction Model for Bollywood Movie Success: A Comparative Analysis of Performance of Supervised Machine Learning Algorithms," The Review of Socionetwork Strategies, 2019.

[9] S. Mundra, A. Dhingra, A. Kapur and D. Joshi, "Prediction of a Movie’s Success Using," in Information and Communication Technology for Intelligent Systems, Singapore, 2019.

[10] R. Parimi and D. Caragea, "Pre-release Box-Office Success Prediction for Motion Pictures," Machine Learning and Data Mining in Pattern Recognition, vol. 7988, pp. 571-585, 2013.

[11] K. Lee, J. Park, I. Kim and Y. Choi, "Predicting movie success with machine learning techniques: ways to improve accuracy," Information Systems Frontiers, vol. 20, pp. 577-588, 2018.

[12] S. Anunaya, "Data Preprocessing in Data Mining -A Hands On Guide," 10 08 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/datapreprocessing-in-data-mining-a-hands-on-guide/. [Accessed 12 08 2021].

[13] J. Browniee, "How to Perform Feature Selection With Numerical Input Data," 18 08 2020. [Online]. Available: https://machinelearningmastery.com/feature-selectionwith-numerical-input-data/. [Accessed 21 05 2021].

[14] J. Brownlee, "A Gentle Introduction to the Chi-Squared Test for Machine Learning," 31 10 2019. [Online]. Available: https://machinelearningmastery.com/chisquared-test-for-machine-learning/. [Accessed 21 05 2021]

[15] J. Brownlee, "How to Choose a Feature Selection Method For Machine Learning," 27 11 2019. [Online]. Available: https://machinelearningmastery.com/feature-selectionwith-real-and-categorical-data/. [Accessed 19 03 2021].