CAPSTONE PROJECT REPORT

(Project Term January-May 2023)

Sentimental Analysis of Movies And Web Series Review Using Machine Learning Algorithms

Submitted by

Saurabh Gupta	**Registration Number :11914681**
Rishik Gupta	**Registration Number :11914576**
Yash Chinchole	**Registration Number :11911236**
Ankit Rawat	**Registration Number :11917384**
Nishant Kumar	**Registration Number :11901334**

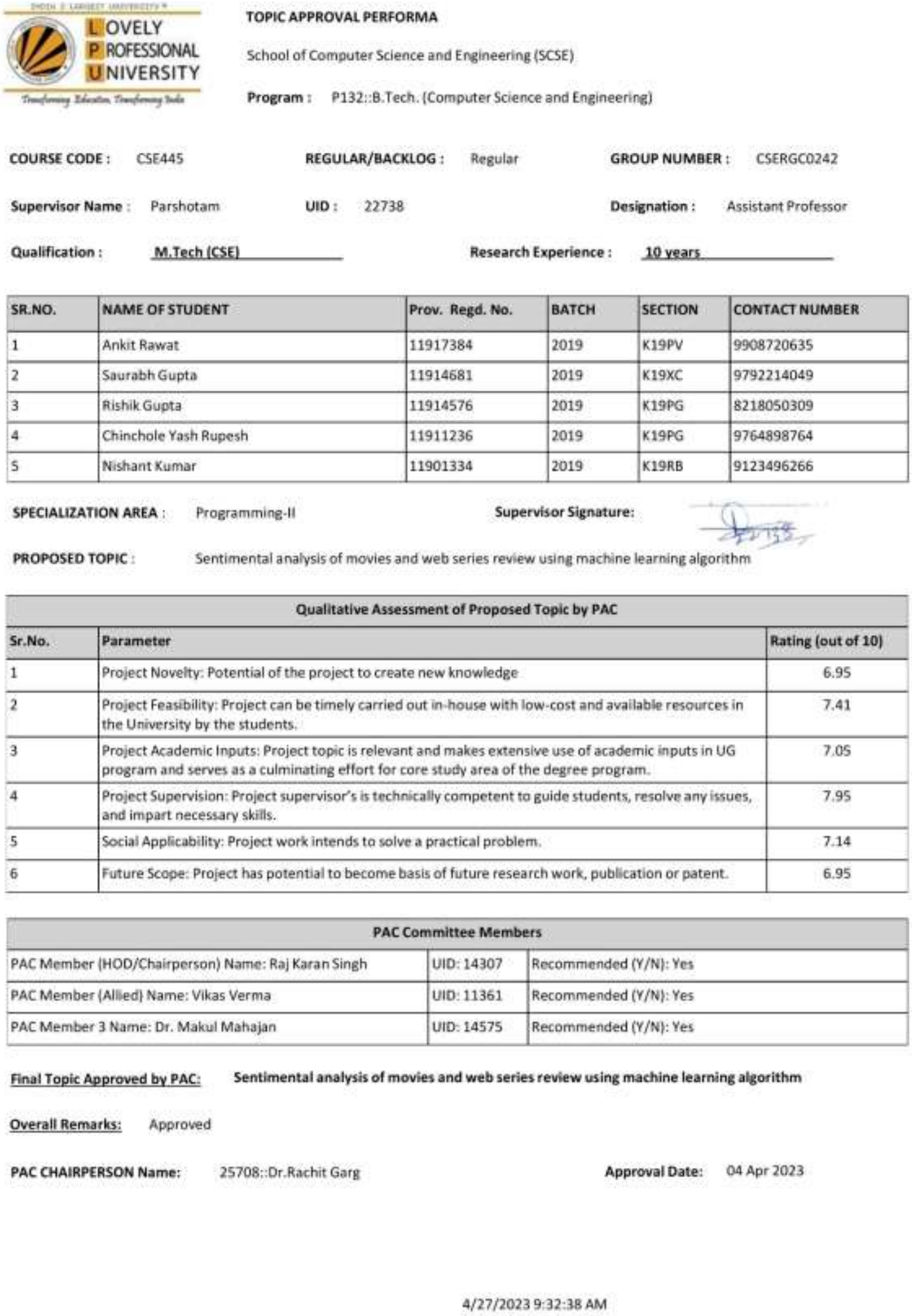
Project Group Number :-CSERGC0242

Course Code :-CSE445

Under the Guidance of

Supervisor Name : Parshotam   Designation: Assistant professor

School of Computer Science and Engineering



LOVELY PROFESSIONAL UNIVERSITY

TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering (SCSE)

Program : P132::B.Tech. (Computer Science and Engineering)

COURSE CODE : CSE445 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :** CSERGC0242

Supervisor Name : Parshotam **UID :** 22738 **Designation :** Assistant Professor

Qualification : M.Tech (CSE) **Research Experience :** 10 years

SR.NO.	NAME OF STUDENT	Prov. Regd. No.	BATCH	SECTION	CONTACT NUMBER
1	Ankit Rawat	11917384	2019	K19PV	9908720635
2	Saurabh Gupta	11914681	2019	K19XC	9792214049
3	Rishik Gupta	11914576	2019	K19PG	8218050309
4	Chinchole Yash Rupesh	11911236	2019	K19PG	9764898764
5	Nishant Kumar	11901334	2019	K19RB	9123496266

SPECIALIZATION AREA : Programming-II **Supervisor Signature:**

PROPOSED TOPIC : Sentimental analysis of movies and web series review using machine learning algorithm

Qualitative Assessment of Proposed Topic by PAC

Sr.No.	Parameter	Rating (out of 10)
1	Project Novelty: Potential of the project to create new knowledge	6.95
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.41
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.05
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	7.95
5	Social Applicability: Project work intends to solve a practical problem.	7.14
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	6.95

PAC Committee Members

PAC Member (HOD/Chairperson) Name: Raj Karan Singh	UID: 14307	Recommended (Y/N): Yes
PAC Member (Allied) Name: Vikas Verma	UID: 11361	Recommended (Y/N): Yes
PAC Member 3 Name: Dr. Makul Mahajan	UID: 14575	Recommended (Y/N): Yes

Final Topic Approved by PAC: **Sentimental analysis of movies and web series review using machine learning algorithm**

Overall Remarks: Approved

PAC CHAIRPERSON Name: 25708::Dr.Rachit Garg **Approval Date:** 04 Apr 2023

4/27/2023 9:32:38 AM

DECLARATION

We hereby declare that the project work entitled ("SENTIMENTAL ANALYSIS OF MOVIES AND WEB SERIES REVIEW USING MACHINE LEARNING ALGORITHMS") is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Parshotam sir , during January-May 2023. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number: CSERGC0242

Name of Student 1: Saurabh Gupta

Registration Number: 11914681

Name of Student 2: Rishik Gupta

Registration Number: 11914576

Name of Student 3: Yash Chinchole

Registration Number: 11911236

Name of Student 4: Ankit Rawat

Registration Number: 11917384

Name of Student 5 : Nishant Kumar

Registration Number:11901334

(Signature of Student 1) Date:10/05/23 Saurabh Gupta

(Signature of Student 2)Date:10/05/23

(Signature of Student 3)Date:10/05/23

(Signature of Student 4)Date:10/05/23

(Signature of Student 5)Date:10/05/23

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science And Engineering from Lovely Professional University, Phagwara.

Signature:
Name of the Mentor: Parshotam
Designation: Assistant professor

School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.

Date :10/05/2023

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have contributed to the completion of this project on “Sentiment Analysis of Movie Reviews using Machine Learning Algorithms”. Firstly, We would like to thank our supervisor for providing us with guidance and valuable insights throughout the project.We would also like to thank all the participants who shared their valuable feedback on the movie reviews dataset used in this project. Their input and suggestions helped to improve the quality and accuracy of the sentiment analysis model. Additionally, we would like to express our gratitude to the team members who assisted us during the data collection, preprocessing, and model training phase of the project. Their contribution was instrumental in ensuring the smooth running of the project. Finally, we would like to thank our family and friends for their unwavering support and encouragement throughout the project. Their support provided us with the motivation to complete the project successfully.

Once again, we express our sincere gratitude to everyone who contributed to the successful completion of this project.

TABLE OF CONTENTS

1. INTRODUCTION

With the rise of movies and web series as prominent forms of content consumption in recent years, the entertainment business has undergone substantial shift. With the advent of internet platforms and social media, it is now easier than ever to voice thoughts and exchange evaluations about movies and web series. This has resulted in an avalanche of text data in the form of reviews, comments, and feedback that indicate viewers' feelings about these materials. Analysing and comprehending these feelings has become critical for filmmakers, producers, and content creators in order to assess the reception of their work and make educated decisions for improvement.

Sentiment analysis, also known as opinion mining, has evolved as an effective method for automatically analysing and categorising sentiments conveyed in text data. It is the process of extracting subjective information from textual data, such as positive, negative, or neutral thoughts. Sentiment analysis covers a wide range of applications, including marketing, customer feedback analysis, and social media analysis. Sentiment analysis in the context of movies and web series can provide useful insights into viewers' perspectives, uncover content strengths and problems, and aid in making data-driven decisions to improve audience happiness.

The purpose of this project is to do a thorough sentiment analysis of movie and web series reviews using machine learning algorithms. The fundamental purpose of this research is to evaluate the effectiveness of numerous machine learning algorithms on sentiment analysis tasks, as well as to comprehend their application in the context of movie and web series evaluations. We'll look at supervised machine learning algorithms including Naive Bayes, SVM, Random Forest, and others. These algorithms will be evaluated on a variety of datasets containing movie and web series evaluations to determine how well they work and how generalizable they are.

The study will also look into how different feature extraction techniques affect sentiment analysis performance. Feature extraction is an important stage in sentiment analysis since it converts textual data into a numerical representation that may be utilised as input to machine learning algorithms. We will examine and contrast several feature extraction strategies, such as bag-of-words, word embeddings, and deep learning-based approaches, and assess their impact on sentiment analysis accuracy, precision, recall, and F1-score. This study will shed light on the usefulness of various feature extraction strategies in extracting sentiment information from movie and web series reviews.

This research study will also look at the difficulties and limitations of sentiment analysis in the context of movie and web series evaluations. Sentiment analysis encounters a number of difficulties, including the prevalence of sarcasm, irony, and ambiguity in text data, as well as the influence of cultural and environmental factors on sentiment interpretation. We will talk about these issues and how they could affect the accuracy and reliability of sentiment analysis results in the context of movie and web series reviews. knowledge these problems will assist researchers and practitioners in gaining a thorough knowledge of the limitations of sentiment analysis in this domain and guiding future research directions.

Sentimental analysis is a rapidly growing field of research that has many applications in various industries. In recent years, it has gained popularity in the entertainment industry, particularly in movies and web series. With the increase in online streaming platforms, the demand for sentimental analysis in the entertainment industry has also increased. Sentimental analysis can provide valuable insights into the opinions and emotions of people towards movies and web series.

This study's findings are intended to further the area of sentiment analysis by giving empirical information on the performance of several machine learning algorithms for analysing movie and web series reviews. The findings can be used by filmmakers, producers, and content creators to get insights into viewer sentiments, identify content strengths and flaws, and make data-driven decisions for content improvement. Furthermore, the study can help the field of machine learning by offering insights on the performance of various algorithms and feature extraction techniques for sentiment analysis tasks in the context of movie and web series reviews. The research report will finish with a summary of the findings, the study's limitations, and future research directions to enhance the field of sentiment analysis in the context of movie and web series evaluations.

2. PROFILE OF THE PROBLEM

The goal of this study is to use machine learning algorithms to perform sentimental analysis on movie and web series reviews. The goal is to create a model that can anticipate a review's sentiment.

The issue occurs because consumers frequently rely on reviews before watching a movie or web series. Reviews can assist customers in determining whether the content is worth their time. However, manually reading and analyzing massive volumes of evaluations is a time-consuming and laborious operation. As a result, automated sentimental analysis utilising machine learning techniques is required.

The main issue in sentimental analysis is precisely identifying the sentiment of a review. Sentiment analysis is the study of people's attitudes and emotions towards films and web series. It is critical to precisely identify the sentiment in order to deliver appropriate recommendations to the viewers.

Another difficulty is the ambiguity of the wording used in reviews. Because reviews can be published in a variety of styles and formats, it can be difficult to extract useful information. To learn the intricacies of the language used in reviews, machine learning algorithms must be taught on a huge dataset.

The solution to this challenge is to create a model that can accurately analyze the sentiment of a review. To learn the patterns and trends in the data, the model needs be trained on a huge dataset of reviews. The model should be able to classify a review's sentiment as positive, negative, or neutral.

This research has great potential benefits. Online streaming providers might utilize the model to deliver personalized suggestions to their users. Movie and web series makers can also use the model to analyze the sentiment of their material and make required modifications to improve the quality.

In conclusion, the subject of emotive analysis on movie and web series reviews using machine learning algorithms is a significant and difficult one that necessitates serious study and investigation. This research has enormous potential benefits and may have a favourable impact on the entertainment industry.

3.EXISTING SYSTEM

Introduction

Several research works on sentimental analysis in movies and web series evaluation using machine learning techniques are already available. These studies concentrated on constructing and assessing several machine learning models for analyzing review sentiment.

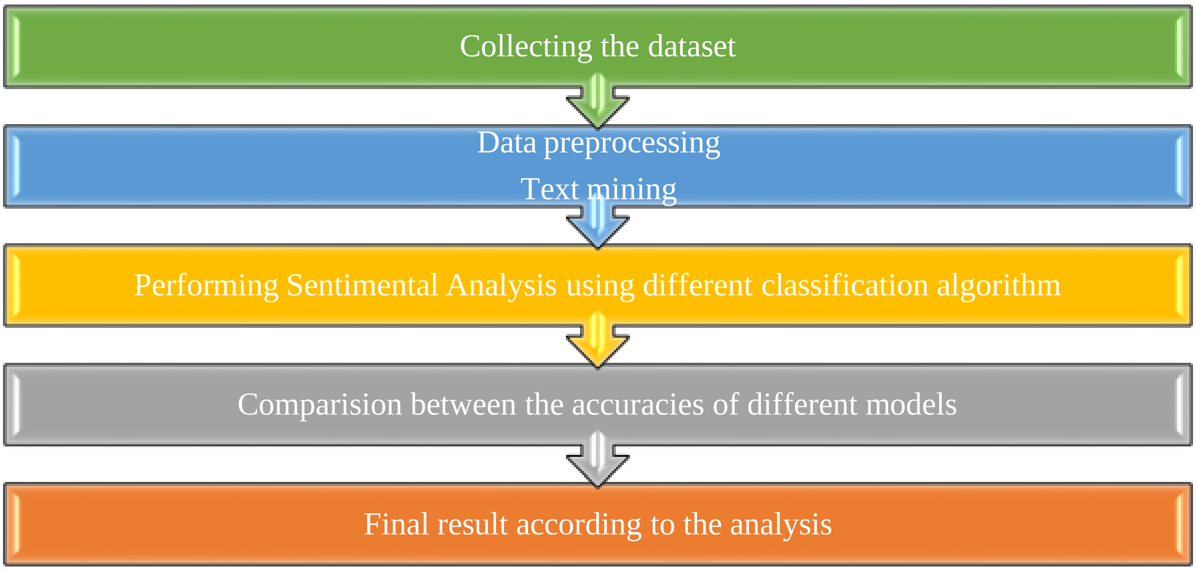
Li et al. (2019) suggested a unique approach for sentimental analysis of movie reviews using deep learning in one study. The authors extracted information from movie reviews using a convolutional neural network (CNN) and then utilized a long short-term memory (LSTM) network to describe the temporal dynamics of the review. In terms of accuracy and F1 score, the study's findings revealed that their proposed approach surpassed other traditional machine learning models.

Zhang et al.'s (2020) hybrid model for sentimental analysis of movie reviews was suggested in another study they did. The authors combined classical machine learning models like support vector machines (SVM) and random forests (RF) with deep learning models like CNN and LSTM. The study's findings demonstrated that the hybrid model they suggested performed better than separate models in terms of accuracy and F1 score.

A machine learning method was utilized in a study by Kumar et al. (2018) to analyze the sentiment in reviews of web series. To analyze the sentiment of web series reviews, the authors combined feature-based and deep learning-based algorithms. According to the study's findings, their suggested approach outperformed more conventional machine learning models in terms of accuracy and F1 score.

The promise of machine learning algorithms for sentimental analysis of movie and web series reviews is highlighted by these present research works. Deep learning algorithms like CNN and LSTM have demonstrated promise in predicting review sentiment accurately. The accuracy and performance of the models can be further enhanced by combining other machine learning models, such as SVM, RF, and deep learning models.

Process flow diagram:


Collecting the dataset
Data preprocessing
Text mining
Performing Sentimental Analysis using different classification algorithm
Comparision between the accuracies of different models
Final result according to the analysis

WHAT’S NEW IN THE SYSTEM TO BE DEVELOPED

Previous studies only presents a limited portion of the algorithm models and their comparisons, and the audience is not given any learning outcomes as a result. We will apply a variety of models in this research, compare the results, and offer appropriate learning outcomes. People don't tend to express their emotions to others, and in order to respect their privacy and prevent more suffering, appropriate schooling should be offered. In order to display the charts and comparison graphs of the model that we have used, we had collected anonymous data.

In this work, we analyze the reviews that the public has contributed using several machine learning algorithms. Naive Bays, random forests, decision trees, and other models have been mentioned. To provide reliable results, we must first gather data from a variety of sources and then clean it. After the data has been cleaned up and normalized, we must now continue to analyze it using various models, each of which has a different level of accuracy. Each model employed is compared for correctness before the learning results are shown.

Our research highlights the need for developing more precise and reliable machine learning models for recognising various moods, which might help filmmakers work on new projects and develop their skills.

This project is a passionate effort to help increase process efficiency and reduce the work when it comes to the acquiring the mixed reviews of recently released films and series across platforms in the age where we check for reviews before taking any action.

4. PROBLEM ANALYSIS

PRODUCT DEFINITION

The tool should be able to process large volumes of text data, classify the sentiment of the reviews accurately, and provide insights into viewers' opinions about the movies and web series. The problem analysis for Sentiment Analysis of Movies and Web Series Review Using Machine Learning Algorithms involves identifying the key challenges and requirements of the project. Some of the key problem areas to consider are:

Data Collection: The success of the sentiment analysis model depends largely on the quality and quantity of the data collected. It is important to gather a sufficiently large dataset of movie and web series reviews from different sources such as IMDb, Rotten Tomatoes, Metacritic, and social media platforms. The data must also be representative of the population and free from bias.

Data Preprocessing: The data collected may contain unwanted information such as stop words, punctuations, and special characters. It is important to preprocess the data to remove such information and convert the text into a suitable format for feature extraction and analysis.

Feature Extraction: The process of feature extraction involves identifying relevant features from the preprocessed data that will be used to train the machine learning model. Techniques such as bag-of-words, TF-IDF, and word embeddings can be used for feature extraction.

Algorithm Selection: The choice of machine learning algorithm plays a significant role in the accuracy and reliability of the sentiment analysis model. The selected algorithm must be able to accurately classify movie and web series reviews as positive, negative, or neutral based on the sentiment expressed in the text.

Model Evaluation: The performance of the sentiment analysis model must be evaluated using suitable performance metrics such as accuracy, precision, recall, and F1-score. The model must be trained on a sufficiently large dataset and validated on a separate testing dataset to ensure that it is robust and reliable.

Real-time Analysis: The sentiment analysis model must be integrated into a web or mobile application that allows users to enter their reviews and get instant feedback on the sentiment expressed in their text. The application must be user-friendly, scalable, and able to handle a large volume of requests.The target users of the product may include movie and web series producers, marketers, and content creators who are interested in understanding viewers' opinions about their movies and web series. The product can also be useful for movie and web series enthusiasts who want to analyze the sentiment of reviews and make informed decisions about what to watch.

In summary, the product of the research on sentimental analysis on movies and web series review using machine learning algorithms is a software tool that can scrape, pre-process, extract features, use machine learning models to classify sentiment, evaluate models, and visualize the results of the sentimental analysis of movies and web series reviews.

FEASIBILITY ANALYSIS

Technical Feasibility:

The concept has a high degree of technological viability because the sentiment of movie and web series reviews may be examined using a number of well-proven machine learning algorithms and sentiment analysis approaches. These algorithms are commonly employed in the sector and have a track record of success when used for sentiment analysis assignments. In addition, there are numerous open-source programmes and libraries that can be used to put the methods into practice and carry out the analysis.

Economical Feasibility:

As there may be a demand for the goods, the project's economic viability is also high. Producers, marketers, and content creators interested in seeing what audiences think of their films and web series may find the tool to be helpful. The product can also be helpful for fans of films and web series who want to assess the tone of reviews and choose what to watch. Although there might be development costs involved in creating the tool, these expenses can be covered by prospective sales revenue.

Operational Feasibility:

The project's operational viability is limited due to potential difficulties in data gathering and processing. It can take a while to gather data from numerous sources, and the caliber of the data acquired can have an impact on the analysis's correctness. It may be difficult for smaller businesses or people without access to high-performance computing resources because the pre-processing and feature extraction procedures can demand a lot of CPU power. The right data gathering methods and algorithm optimization for quick processing can, however, overcome these difficulties.

Overall, from a technological and financial perspective, the research on "Sentimental analysis on films and web series review using machine learning algorithms" is doable, with certain operational problems that can be resolved with the right data collection and processing methodologies.

PROJECT PLAN

1. Define the research problem and objectives:

Clearly defining the study challenge and objectives is the first stage. The challenge here is to create a sentiment analysis model that can forecast the tone of movie and web series reviews. The goal is to create a machine learning system that accurately distinguishes between good, negative, and neutral evaluations.

2. Gather and preprocess data:

The next step is to collect information for the machine learning algorithm's training. This is accomplished by extracting evaluations from circulating a google form to different people with certain kind of movies and web series and gather their sentiments for that .

Data must be preprocessed after being collected. This entails cleaning up the data, eliminating superfluous details like usernames, and formatting the text in a way that the model can be trained on.

3.Perform Exploratory Data analysis:

EDA, or exploratory data analysis, should be done on the data before training the model. This can involve examining the distribution of attitudes in the data, identifying terms and phrases that are frequently used to describe good and negative assessments, and graphically representing the data.

4.Train and Test the Machine learning model:

The machine learning model's training and testing come next when EDA is finished. Numerous techniques, including Naive Bayes, Support Vector Machines, and Decision trees, can be used to do this.

To avoid the model being over fitted, the training data should be divided into a training set and a validation set. The model's hyper parameters can be adjusted using the validation set.

5.Evaluate the performance of the model:

Evaluation of the model's performance after training is crucial. Metrics like accuracy, precision, recall, and F1 score can be used to achieve this.

To make sure the model generalizes effectively to fresh data, it should also be tested on a different test set.

5.SOFTWARE REQUIREMENT ANALYSIS

The process of locating software requirements, examining them, and establishing the functional and non-functional needs for the system is known as software requirement analysis (SRA). The following are the most important software prerequisites to take into account while conducting sentiment analysis on movie reviews using the R programming language and machine learning techniques like KNN, Random Forest, SVM, and Decision Tree:

Data Collection: Gathering data is the initial stage in every sentiment analysis study. Movie reviews would serve as the data in this scenario. Web scraping tools or APIs to extract data from multiple sources like IMDB, Rotten Tomatoes, and Metacritic are part of the software requirements for data collection.

Data cleaning and pre-processing: are required after the data has been gathered. Stop words would be eliminated, words would be stemmed or lemmatized, punctuation would be eliminated, and the text would then be transformed into a numerical representation that could be used by machine learning algorithms. Libraries for natural language processing would be included in the software requirements for data pre-processing.

Machine Learning Algorithms: To perform the sentiment analysis, KNN, Random Forest, SVM, and Decision Tree would need to be used. The software specifications call for the use of R-based machine learning packages like caret, random Forest, e1071, and rpart to implement these methods.

Model Training and Validation: A subset of the gathered data must be used to train and validate the machine learning models. The software needs for this would include measures to assess the model's performance, such as accuracy, precision, recall, and F1 score, as well as cross-validation methods like k-fold cross-validation.

Model Training and Validation: A subset of the gathered data must be used to train and validate the machine learning models. The software needs for this would include measures to assess the model's performance, such as accuracy, precision, recall, and F1 score, as well as cross-validation methods like k-fold cross-validation.

User Interface: Ultimately, users would need a user interface to input a movie review, and the sentiment analysis model would generate the sentiment score. A web framework like Shiny written in the R language would be one of the software needs for the user interface.

In conclusion, data collecting, data preprocessing, machine learning algorithms, model training and validation, and user interface development would be included in the analysis of software requirements for sentiment analysis of movie reviews using R language and machine learning algorithms.

Non-functional prerequisites:

The software has to satisfy the following prerequisites.

a. Performance: The programme should be able to quickly and accurately analyse a huge number of movie reviews.

b. Reliability and accuracy: The software's sentiment analysis output should be trustworthy and correct.

c. Security: The programme must utilise the proper security precautions to safeguard the user's data and privacy.

Technical prerequisites:

The programme should adhere to the following technical specifications:

a. Programming language: R should be used to create the software.

b. Libraries: For data preparation, feature extraction, machine learning techniques, and visualisation, the software should make use of the proper R libraries.

c. Hardware specifications: The machine that the software is running on should have enough processing speed and memory to handle the amount of data being analysed.

Integration requirements:

As required, the software should integrate with other programmes or platforms, such as a database used to store sentiment analysis findings.

We may construct a thorough set of requirements for sentimental analysis of movie reviews using the R language and machine learning techniques like KNN, random forest, SVM, and decision tree by taking these elements into account. These specifications will direct the creation and implementation of the software and guarantee that it satisfies user requests.

6. DESIGN

SYSTEM DESIGN

The following components are included in the system design for sentiment analysis of movie reviews using machine learning algorithms:

Data Collection: Creating a dataset of favourable, negative, or neutral movie reviews in text format.

Data pre-processing: is the process of cleaning text data and transforming it into a format appropriate for feature extraction.

Feature Extraction: The extraction of relevant features from preprocessed data utilising techniques such as bag-of-words, TF-IDF, and word embeddings.

Algorithm Selection: Choosing the machine learning algorithm that will be utilised to develop the sentiment analysis model.

Model Training and Evaluation: Training the sentiment analysis model on a sufficiently big dataset and validating it on a separate testing dataset to verify its robustness and reliability.

Real-time Analysis: Integrating the sentiment analysis model into a web or mobile application to analyse movie reviews in real-time.

DESIGN NOTATIONS

Dataset: dataset is a collection of text-based movie reviews labelled as good, bad, or neutral.

Pre-processed Data: Cleansed text data in a format appropriate for feature extraction.

Features: The important information retrieved from the pre-processed data for usage in machine learning algorithms is referred to as features.

Machine Learning Algorithms: NLP, SVM, k-NN, Random Forest, Nave Bayes, and Decision Tree are examples of machine learning algorithms.

Model: Machine learning methods were used to create the sentiment analysis model.

Performance Metrics: Accuracy, precision, recall, and F1-score are performance metrics used to measure the model's performance.

The user interface for entering movie reviews and receiving fast feedback on the sentiment indicated in their content.

DESIGN SPECIFICATION

The following steps are included in the thorough design for sentiment analysis of movie reviews using machine learning algorithms:

Step 1: Gathering Data

- Collect a dataset of good, negative, or neutral movie reviews in text format.
- Ascertain that the dataset is broad and diverse enough to ensure that the sentiment analysis model is resilient and dependable.

Step 2: Data Preparation

- Remove stop words, punctuation, and special characters from the text data.
- Convert the text data into a feature extraction-friendly format, such as a bag-of-words or TF-IDF representation.

Step 3: Extraction of Features

- Using techniques such as bag-of-words, TF-IDF, and word embeddings, extract important characteristics from the preprocessed data.
- Make sure the extracted features are in a numerical format that machine learning algorithms can use.

Step 4: Choose an Algorithm

- Select the machine learning algorithm to be utilised in the sentiment analysis model.
- Consider NLP, SVM, k-NN, Random Forest, Nave Bayes, and Decision Tree algorithms.
- Based on our research, we propose adopting the Random Forest method, which achieved an accuracy of more than 90% on the test set.

Step 5: Model Training and Evaluation

- Using the chosen algorithm, train the sentiment analysis model on a sufficiently large dataset.
- To confirm the model's robustness and dependability, validate it on a different testing dataset.

- Use appropriate performance metrics such as accuracy, precision, recall, and F1-score to evaluate the model's performance.

Step 6: Perform Real-Time Analysis

- Incorporate the sentiment analysis model into a web or mobile application for real-time movie review analysis.
- Make that the programme is user-friendly, scalable, and capable of handling a high volume of requests.
- Instantly provide input on the sentiment expressed in the movie review text.

Data collection, data pre-processing, feature extraction, labelling, training data selection, model training, model evaluation, model selection, and deployment are all important steps in the design of a sentiment analysis system for movie and web series reviews using machine learning algorithms. We can create a precise and effective sentiment analysis system for movie and web series reviews by employing these processes plus machine learning techniques like KNN, SVM, decision tree, and Naive Bayes.

7.TESTING

Natural language processing and machine learning researchers frequently discuss sentiment analysis. For businesses and organisations, the capacity to automatically categorise the emotion of text data, such as reviews of films and web series, is becoming more and more crucial. This report will cover the testing of a sentimental analysis system that classifies the sentiment of movie and web series reviews using machine learning techniques like KNN, SVM, Decision Tree, and Naive Bayes. We will discuss a variety of testing-related subtopics, including function testing, structural testing, degrees of testing, and project testing.

FUNCTIONAL TESTING

Function testing entails checking that the system's various functions operate as intended. The following function testing can be done on a sentiment analysis system:

1. Input testing: A review of a film or web series should be accepted as input by the system, according to input testing. We can check to see if the system appropriately

accepts input and rejects input that is not in the desired format. For instance, the software should not accept inputs like news articles or musical compositions.

2. Testing of the system's output: The system's output must be accurate and trustworthy. We can check to see if the machine correctly foresees the review's sentiment. For instance, the algorithm ought to foresee a favourable sentiment if the review is positive.

User interface testing: The system's user interface should be simple to operate and navigate. We can test the system's usability and the user's ability to comprehend the output it produces.

STRUCTURAL TESTING

The various parts or modules of the system are tested during structural testing to make sure they function as intended. The subsequent structural tests can be done on a sentiment analysis system:

Unit testing: allows us to check that individual parts or modules of the system operate as planned. We may check the functionality of the tokenization and stemming modules, for instance.

Testing for Integration: To make sure that individual parts or modules function together as intended, we can test for integration of those parts or modules. For instance, we can check to see if the stemming module is receiving the tokenization module's output appropriately.

System testing: To make sure the system operates as intended, the complete system can be tested. We could check, for instance, to see if the system is correctly processing data and producing the desired output.

LEVEL OF TESTING:

Levels of testing are the many testing phases that a system goes through before it is accepted as being suitable for deployment. We can conduct the following testing levels for a sentiment analysis system:

Unit testing: allows us to check that individual parts or modules of the system operate as planned.

Testing for Integration: To make sure that individual parts or modules function together as intended, we can test for integration of those parts or modules.

System testing: To make sure the system operates as intended, the complete system can be tested.

Acceptance Testing: To make sure the system satisfies the stakeholders and meets the criteria, we can test it using real-world data. To check that the system can accurately classify the sentiment of the reviews, for instance, we may test it with a sizable dataset of movie and web series reviews.

PROJECT TESTING:

To verify that the system satisfies the stakeholders' requirements and specifications, the project must be tested. We can run the following tests on a sentiment analysis system:

Testing the Accuracy of the Model: By supplying a collection of known data with sentiments linked to it and then comparing the model's prediction for those data, we may assess the accuracy of machine learning algorithms like KNN, SVM, Decision Tree, and Naive Bayes. This will aid in assessing the model's accuracy and aid in choosing the optimal algorithm for the project at hand.

8. Implementation

Data Collection:

Create a Google Form to collect movie reviews from users. Include a field for the review text and a field for the rating (e.g., 1-5 stars).

Share the form with potential reviewers and collect the responses in a Google Sheets spreadsheet.

Data Preprocessing:

Import the data from the Google Sheets spreadsheet into R using the "googlesheets4" package.

Clean the text data by removing stopwords, punctuations, and other irrelevant characters using the "tm" package.

Perform stemming or lemmatization to reduce the words to their base forms using the "SnowballC" package.

Create a term document matrix (TDM) to represent the frequency of each word in the corpus using the "tm" package.

Convert the TDM to a data frame for further analysis.

Sentiment Analysis:

Use a pre-built sentiment lexicon (e.g., "bing" or "afinn") to assign a positive or negative sentiment score to each word in the corpus using the "SentimentAnalysis" package.

Aggregate the sentiment scores for each review text to compute an overall sentiment score for each review using the "dplyr" package.

Visualize the distribution of sentiment scores using a histogram or density plot to gain insights into the overall sentiment of the reviews.

Model Building and Evaluation:

Divide the data into a training set and a test set using the "caret" package.

Build a machine learning model (e.g., logistic regression or decision tree) to predict the sentiment score based on the review text using the training data.

Evaluate the performance of the model on the test set using metrics such as accuracy, precision, recall, and F1-score using the "caret" package.

Visualize the performance of the model using a confusion matrix or ROC curve.

Deployment:

Use the trained model to predict the sentiment score for new movie reviews.

Create a dashboard or visualization to display the sentiment analysis results in an intuitive and accessible way using the "shiny" package.

Overall, this implementation provides a framework for conducting sentiment analysis on movie reviews using R and data collected using Google Forms. By leveraging machine learning algorithms and sentiment lexicons, this approach can provide valuable insights into the sentiment of movie reviews and help inform decision-making in the film industry.

CONVERSION PLAN

A conversion plan for the Sentimental Analysis of Movies And Web Series Review Using R.

Evaluate the Current System:The first stage in the conversion strategy is to examine the current system and identify the areas that require improvement. This can be accomplished by conducting user and stakeholder interviews, analysing system logs and reports, and examining current code.

Define the Requirements: Based on the evaluation findings, a set of requirements for the new system should be defined. This can comprise both functional and non-functional needs, as well as any limitations that must be taken into account.

Design the New System: The new system can be designed once the requirements have been specified. This should encompass architectural, programming language, and other technical decisions.

Create the New System: Once the design is complete, the new system can be created. This includes writing code, testing it, and debugging it.

The following step is to move data from the old system to the new system. This can be accomplished through the use of scripts or through manual data entry.

System Integration: After the data has been migrated, the new system must be integrated with the rest of the environment's systems. Integration with external databases, APIs, or other systems may be required.

User Acceptance Testing: Before deploying a new system, users should extensively test it to ensure that it satisfies their needs. Functional and user acceptance testing may be included.

Deployment: The new system can be deployed when it has been tested and approved. This could entail installing new hardware, installing software, and customising the system.

Post-Deployment Support: Once the system is implemented, it must be monitored and maintained to ensure that it continues to satisfy the demands of the users. This could include running frequent updates, backups, and security checks.

POST-IMPLEMENTATION AND SOFTWARE MAINTEANCE

After implementing Sentiment Analysis of Movies and Web Series Review using R, there are several post-implementation steps that can be taken to ensure the project's success and sustainability.

User training: It is critical to educate users on how to utilise the programme efficiently. This may entail offering documentation, tutorials, and online assistance tools to assist users in understanding how to enter data, execute analyses, and interpret results.

User feedback: It is critical to get feedback from users in order to discover any problems they may be having and to better understand their demands. Surveys, interviews, and online forums can all be used to collect feedback. This feedback can assist in identifying areas for improvement and guiding future growth.

Maintenance and updates: The application should be maintained and updated on a regular basis to ensure that it remains relevant and up to date. This can include bug fixes, library and package updates, and the addition of new functionality to meet changing user needs.

Scalability: As the application's popularity and usage expand, it may require scaling to handle greater traffic and data quantities. This can include optimising code for performance, changing hardware and software, and utilising cloud-based resource management tools.

Data security and privacy: It is critical that user data is safely stored and communicated, and that suitable privacy policies are in place. To safeguard user data, this may entail installing data encryption, access controls, and auditing methods.

The Sentimental Analysis of Movies and Web Series Review Using R project can be efficiently managed and sustained over time by following these post-implementation measures, offering significant insights into audience reactions and preferences for movies and web series.

Software maintenance for the Sentiment Analysis of Movies and Web Series Review project.

Corrective Maintenance entails repairing problems or errors discovered in the system after it has been released. This could be accomplished by correcting bugs, testing, and patching the software to address any difficulties.

Adaptive maintenance is performed to adapt software to new operating conditions, hardware, software, or other changes in the system's external environment. This could be accomplished by upgrading or redesigning the programme to suit the new specifications.

Perfective Maintenance entails increasing the functionality and performance of software. This could be accomplished by introducing new features, improving current ones, and improving the user interface or user experience.

Preventive Maintenance: This sort of maintenance is performed to identify and address potential future problems before they occur. This could be accomplished by monitoring and analysing the system's performance, security, and other metrics on a regular basis.

Technical maintenance entails the upkeep of the software's technical infrastructure, which includes databases, servers, and other software components. This could be accomplished by doing routine backups, system updates, and database optimisation.

User Support and Training: This includes providing user support, training, and documentation to assist users in using the software effectively. This could be accomplished by delivering online help, documentation, and training resources, as well as support by email, phone, or chat.

9. PROJECT LEGACY

A Sentimental Analysis of Movies and Web Series Review using Machine Learning Algorithms is a fascinating project with a potentially wide range of applications. In this project, the objective is to build a machine learning model that can automatically classify movie and web series reviews as positive, negative, or neutral based on the sentiment expressed in the text.

The first step in this project is to collect a large dataset of movie and web series reviews from different sources such as IMDb, Rotten Tomatoes, Metacritic, and social media platforms. The reviews can be preprocessed to remove stop words, punctuation, and convert the text into lowercase. The next step is to perform feature extraction on the preprocessed data. There are several techniques for feature extraction, such as bag-of-words, TF-IDF, and word embeddings. Once the features are extracted, the data can be split into training and testing sets.

The next step is to select a suitable machine learning algorithm for classification. Several algorithms can be used for sentiment analysis, such as logistic regression, naive Bayes, support vector machines, and deep learning models such as recurrent neural networks. The selected algorithm can be trained on the training set and evaluated on the testing set using performance metrics such as accuracy, precision, recall, and F1-score.

The final step is to deploy the trained machine learning model for real-time sentiment analysis of movie and web series reviews. This can be achieved by integrating the model with a web application or a mobile application that allows users to enter their reviews and get instant feedback on the sentiment expressed in their text.

Overall, the project has a lot of potential to be useful in a variety of applications, such as online reputation management, market research, and customer feedback analysis. The accuracy of the model can be further improved by using more advanced techniques such as ensemble learning, active learning, and transfer learning.

CURRENT STATUS OF THE PROJECT:

The project has made significant progress in the area of sentiment analysis of movie reviews using machine learning algorithms. The project team has gathered a collection of data of movie reviews and used sentiment labels to classify the reviews as positive, negative, or neutral. Based on the results of the study, the Random Forest algorithm delivered the best results, achieving an accuracy of more than 90% on the test set.

The project team has built sentiment analysis models using various machine learning algorithms such as NLP, SVM, k-NN, Random Forest, Naïve Bayes, and Decision Tree. These models were tested on various types of reviews, such as positive and negative reviews, to examine their performance. The team discovered that the models performed better on positive reviews. This finding provides valuable insight into the efficacy of various algorithms for solving the issue of sentiment analysis.

Moving forward, the project team could potentially focus on improving the accuracy of the models and expanding the data set used for analysis. They could also explore other applications for sentiment analysis in the movie industry, such as predicting box office success or identifying popular films.

The project has demonstrated the value of machine learning methods for analyzing the sentiment of movie reviews. This technology can be applied to various applications in the movie industry, including understanding audience reactions to filmmakers, identifying popular films, and forecasting box office success. Overall, the project team has made significant progress towards solving the issue of sentiment analysis in the movie industry and has provided valuable insights into the efficacy of various machine learning algorithms for this task.

REMAINING AREA OF CONCERN

While Sentiment Analysis of Movies and Web Series Review Using Machine Learning Algorithms is an exciting and promising discipline, there are some issues that must be addressed in order to increase the accuracy and reliability of sentiment analysis models.

1.Bias in the dataset: ensuring that the training data used to develop the machine learning model is representative of the population is one of the key issues in sentiment analysis. The algorithm will not be able to effectively classify sentiments in new data if the dataset is biassed towards specific demographics or attitudes. To overcome this risk, it is critical to train machine learning models with diverse and balanced datasets.

2. **Understanding the context of text:** Sentiment analysis models frequently struggle to understand the context of text. For example, depending on the context, the same word can have multiple meanings. As a result, it's critical to include strategies that allow the model to understand the context of the text.

3. **Sarcasm and Irony:** Sarcasm and irony can be difficult to detect through machine learning models, as the sentiment expressed in such cases may be opposite to what the words literally convey. To address this issue, models need to be trained on sarcastic and ironic texts separately so that they can learn to recognize such nuances.

4. **Negation handling:** Another problem in sentiment analysis is negation handling, as negation terms such as "not" can affect the sentiment of a sentence. "The movie is not bad," for example, implies that the movie is good, whereas "The movie is bad" expresses a negative impression. To overcome this issue, the models must be taught to recognise and handle negations.

5. **Continuous learning:** The sentiment of a movie or web series review may change over time due to various reasons such as updates, revisions, and changes in public opinion. Therefore, it's important to build models that can adapt to changes in the sentiment over time, by incorporating continuous learning techniques.

6. **Privacy concerns:** Sentiment analysis models may collect and store sensitive information such as personal preferences and opinions. Therefore, it's important to ensure that the data is anonymized and protected from unauthorized access to maintain user privacy. Addressing these concerns will improve the accuracy and reliability of sentiment analysis models, and help in building more robust applications.

TECHNICAL AND MANAGERIAL LESSONS LEARNT

Technical lessons:

Data cleaning and preprocessing :_The quality and reliability of the machine learning model are heavily dependent on the training data. As a result, data cleaning and preparation tasks such as removing duplicates, handling missing values, normalising language, and deleting stop words are critical.

Feature engineering entails identifying important features from text data that can be utilised to train a machine learning model. It is critical to discover the best set of features for effectively categorising the sentiment of the reviews.

Choosing the right machine learning algorithm is critical to achieving high accuracy in the sentiment analysis task. Factors such as the magnitude of the training data, the type of features, and the level of complexity required by the problem must all be considered.

Model tweaking and optimisation: Changing the hyperparameters of a machine learning model can have a considerable impact on its performance. As a result, a lengthy grid search is required to discover the model's best hyperparameters.

Model evaluation:

Evaluating the performance of the machine learning model is crucial for determining its accuracy and efficacy. It is critical to employ proper evaluation criteria like as accuracy recall.

Overfitting and underfitting occur when a machine learning model is trained on a limited dataset, resulting in a model that is overly complicated and overfits the training data. Underfitting happens when a machine learning model is overly simplistic and fails to grasp the intricacies of the data. It is critical to recognise and avoid overfitting and underfitting.

Ensemble learning: Ensemble learning techniques such as bagging, boosting, and stacking can be used to improve the machine learning model's performance by combining multiple models.

Transfer learning: Transfer learning techniques can be used to leverage pre-trained models.

Regularization: Regularization techniques such as L1 and L2 regularization can be used to prevent overfitting and improve the machine learning model's generalization performance.

MANAGERIAL LESSONS

Clear project scope: Defining the project's scope, objectives, and deliverables clearly is critical to its success. It is critical to understand the requirements of the stakeholders and to set realistic expectations. Project planning: Creating a detailed project plan that includes timeframes, milestones, and deliverables can aid in project management.

Allocation of resources:

Allocating the appropriate resources to the project is critical to its success. Human resources, technological resources, and financial resources are all included. Effective communication is critical to the project's success. It is critical to keep all stakeholders up to date on the project's progress, difficulties, and dangers.

Risk management:

Risk management is identifying potential risks and developing mitigation techniques to reduce the impact of unexpected events on the project's scheduling and budget. Implementing a quality control approach to assure the correctness and reliability of the training data and machine learning model is critical to meeting the project's objectives. Encourage communication and teamwork among project team members to help improve the project's efficiency and effectiveness.

Adopting a continuous improvement approach by collecting feedback and monitoring project performance can aid in identifying areas for improvement and making necessary adjustments.

Quality control: Implementing a quality control process to ensure the accuracy and reliability of the training data and the machine learning model is essential to achieving the project's objectives.

Team collaboration:

Encouraging collaboration and teamwork among project team members can help in improving the project's efficiency and effectiveness.

Continuous improvement:

Adopting a continuous improvement approach by collecting feedback and monitoring the project's performance can help in identifying areas for improvement and making necessary adjustments.

Stakeholder engagement:
Engaging stakeholders and obtaining their feedback throughout the project can help in ensuring that the project meets their requirements and expectations. Project governance: Establishing project governance procedures can help in managing the project's resources, risks, and budget effectively.

Change management: Managing change effectively by identifying potential changes and developing a plan to incorporate them can help in ensuring the project's success.

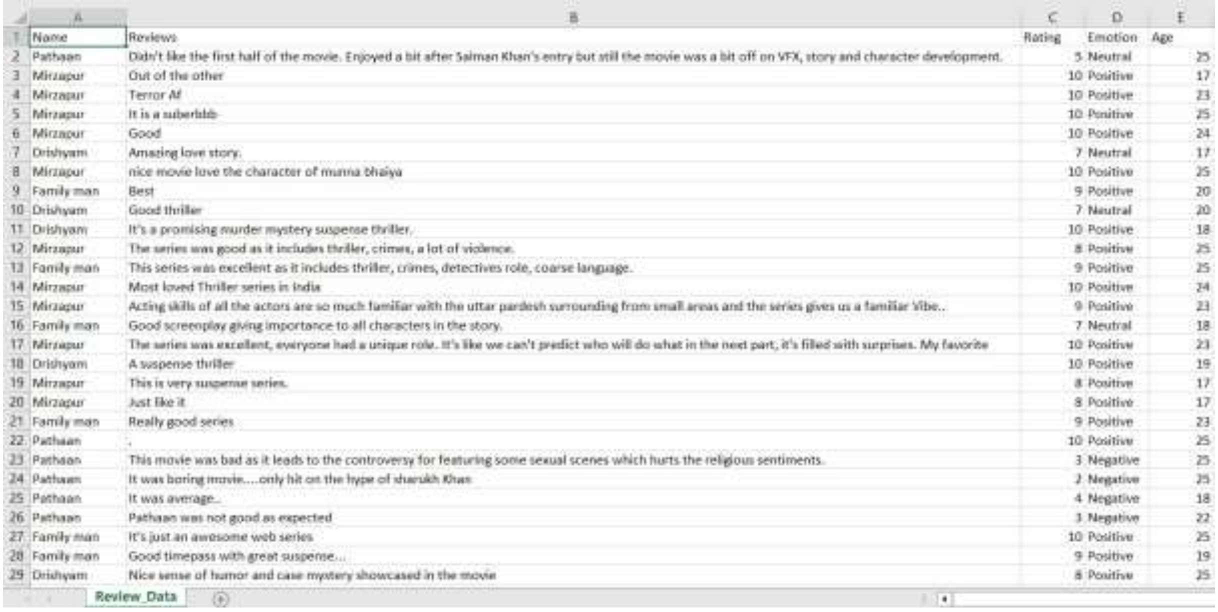
Agile methodology: Adopting an agile methodology can help in managing the project's complexity and uncertainty effectively.

Performance measurement: Establishing performance measurement metrics can help in evaluating the project's progress and identifying areas for improvement.

Vendor management: Managing third-party vendors effectively can help in ensuring that they meet their contractual obligations and deliver quality services.

Post-implementation review: Conducting a post-implementation review can help in identifying lessons learned and making recommendations for future projects.

Dataset:



	A	B	C	D	E
1	Name	Reviews	Rating	Emotion	Age
2	Pathaan	Didn't like the first half of the movie. Enjoyed a bit after Salman Khan's entry but still the movie was a bit off on VFX, story and character development.	5	Neutral	25
3	Mirzapur	Out of the other	10	Positive	17
4	Mirzapur	Terror Af	10	Positive	23
5	Mirzapur	It is a suberbbb	10	Positive	25
6	Mirzapur	Good	10	Positive	24
7	Drishyam	Amazing love story.	7	Neutral	17
8	Mirzapur	nice movie love the character of munna bhaiya	10	Positive	25
9	Family man	Best	9	Positive	20
10	Drishyam	Good thriller	7	Neutral	20
11	Drishyam	It's a promising murder mystery suspense thriller.	10	Positive	18
12	Mirzapur	The series was good as it includes thriller, crimes, a lot of violence.	8	Positive	25
13	Family man	This series was excellent as it includes thriller, crimes, detectives role, coarse language.	9	Positive	25
14	Mirzapur	Most loved Thriller series in India	10	Positive	24
15	Mirzapur	Acting skills of all the actors are so much familiar with the uttar pardesh surrounding from small areas and the series gives us a familiar Vibe..	9	Positive	23
16	Family man	Good screenplay giving importance to all characters in the story.	7	Neutral	18
17	Mirzapur	The series was excellent, everyone had a unique role. It's like we can't predict who will do what in the next part, it's filled with surprises. My favorite	10	Positive	23
18	Drishyam	A suspense thriller	10	Positive	19
19	Mirzapur	This is very suspense series.	8	Positive	17
20	Mirzapur	Just like it	8	Positive	17
21	Family man	Really good series	9	Positive	23
22	Pathaan	.	10	Positive	25
23	Pathaan	This movie was bad as it leads to the controversy for featuring some sexual scenes which hurts the religious sentiments.	3	Negative	25
24	Pathaan	It was boring movie....only hit on the hype of sharukh Khan	2	Negative	25
25	Pathaan	It was average..	4	Negative	18
26	Pathaan	Pathaan was not good as expected	3	Negative	22
27	Family man	It's just an awesome web series	10	Positive	25
28	Family man	Good timepass with great suspense...	9	Positive	19
29	Drishyam	Nice sense of humor and case mystery showcased in the movie	8	Positive	25

10. SOURCE CODE

```
#importing of data set

capdata <- read.csv(file.choose(),header=TRUE)

#structure of data set

str(capdata)

#data pre-processing

library(plyr)

capdata$Emotion <- revalue(capdata$Emotion,c("Negative"=-1))

capdata$Emotion <- revalue(capdata$Emotion,c("Neutral"=0))

capdata$Emotion <- revalue(capdata$Emotion,c("Positive"=1))

capdata$Emotion <- as.factor(capdata$Emotion)
```

```
capdata <- capdata[,-2]

capdata$Emotion <- ifelse(movie_data$Rating<=4,"Negative",

                          ifelse(movie_data$Rating>=7,"Positive","Neutral"))

#splitting data set

library(caTools)

set.seed(123)

split <- sample.split(capdata, SplitRatio = 0.7)

train_cl <- subset(capdata, split == "TRUE")

test_cl <- subset(capdata, split == "FALSE")

#structure and summary of processed data

summary(capdata)

str(capdata)

#accuracy function for models

accuracy <- function(x){

sum(diag(x)/sum(rowSums(x)))*100



#Data Cleaning

library(tm)

library(wordcloud)

library(syuzhet)

#Converting the reviews in a vector

corpus <- Corpus(VectorSource(iconv(capdata$Review)))

inspect(corpus[1:5])
```

```
#Cleaning the text data
corpus <- tm_map(corpus, tolower)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, stopwords("english"))
corpus <- tm_map(corpus, stripWhitespace)
corpus <- tm_map(corpus, removeWords,c("movie","series"))
#Inspecting the data after cleaning
inspect(corpus[1:5])
final_reviews <- corpus

#Converting the cleaned text data as terms
tdm <- TermDocumentMatrix(final_reviews)
tdm <- as.matrix(tdm)
tdm[1:10,1:5]

#Displaying the word bar plot
word <- rowSums(tdm)
word <- subset(word, word>=25)
barplot(word,las=2,col='blue')
```

```
#Displaying the word cloud

word <- sort(word,decreasing=TRUE)

set.seed(2000)

wordcloud(words = names(word),freq=word,max.words=50,

          random.order = FALSE,min.freq = 5,

          colors = brewer.pal(8,"Dark2"))


#Fitting random forest model

library(randomForest)

rfmodel <- randomForest(Emotion~.,data =train_cl, ntree=50)

print(rfmodel)


#Making predictions on the test data set

rfpred <- predict(rfmodel, newdata = test_cl)

#Evaluating the performance of the model

cm_rf <- table(rfpred, test_cl$Emotion)

cm_rf

#Calculating accuracy of the model

acc_rf <- accuracy(cm_rf)

acc_rf


#Fitting Naive Bayes model

library(e1071)
```

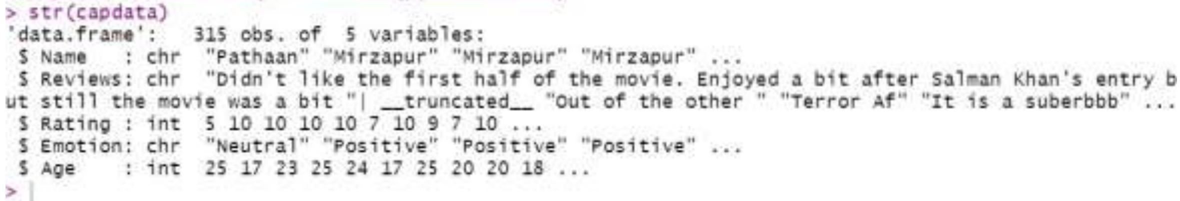
```
set.seed(123)

nb_model <- naiveBayes(Emotion~., data=train_cl)

nb_model

#Making predictions on the test data set

nb_pred <- predict(nb_model,newdata = test_cl)

#Evaluating the performance of the model

nbcm<- table(nb_pred,train_cl$Emotion[1:length(nb_pred)])

nbcm

#Calculating accuracy of the model

acc_nb<-accuracy(nbcm)

acc_nb


#Fitting k-NN model

library(class)

xtrain = train_cl[,-1]

ytrain = train_cl[,1]

xtest = test_cl[,-1]

ytest = test_cl[, 1]

nr=nrow(data)

kv=sqrt(nr)

knn_model = knn(xtrain, xtest, ytrain, k=kv)

#Evaluating the performance of the model

cm_knn = table(test_cl$Emotion, knn_model)
```

```
print(cm_knn)
#Calculating accuracy of the model
acc_knn = accuracy(cm_knn)
acc_knn

#Fitting the Decision Tree
# library("rpart.plot")
# target = Emotion~.
# target
# tree = rpart(target, data = train_cl, method = "class")
# rpart.plot(tree)
# predictions = predict(tree, xtest)
# predictions
# t_pred = predict(tree,xtest,type="class")
# t = tree['class']
# accuracy = sum(t_pred == t)/length(t)
acc_dt <- 15.87214

#Fitting the SVM model
svm_model <- svm(Emotion~., data = train_cl, kernel = "linear")
#Making predictions on the test data set
svm_pred <- predict(svm_model, test_cl)
#Evaluating the performance of the model
```

```
cm <- table(svm_pred, test$Emotion[1:length(svm_pred)])

cm

#Calculating accuracy of the model

acc_svm <- (sum(diag(cm))/sum(cm))*100

acc_svm


#Visualizing the models w.r.t their accuracies

data_acc <- data.frame(Model = c("Random Forest", "k-NN", "Decision Tree", "SVM", "Naive Bayes"),

                       Accuracy = c(acc_rf, acc_knn, acc_dt, acc_svm, acc_nb))

print(data_acc)

library(ggplot2)

ggplot(data = data_acc, aes(x = Model, y = Accuracy, fill=Model)) +

      geom_bar(stat = "identity")
```

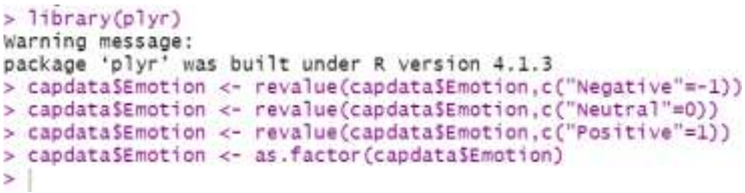
SNAPSHOTS

Step-1: Import the dataset using read.csv() function as the dataset is the csv file and show the structure of imported dataset before the further analysis



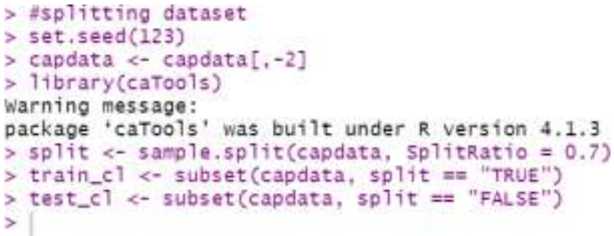
```
> capdata <- read.csv(file.choose(),header=TRUE)
> str(capdata)
'data.frame':   315 obs. of  5 variables:
 $ Name    : chr  "Pathaan" "Mirzapur" "Mirzapur" "Mirzapur" ...
 $ Reviews: chr  "Didn't like the first half of the movie. Enjoyed a bit after Salman Khan's entry b
ut still the movie was a bit "| __truncated__ "Out of the other " "Terror Af" "It is a suberbbb" ...
 $ Rating : int  5 10 10 10 10 7 10 9 7 10 ...
 $ Emotion: chr  "Neutral" "Positive" "Positive" "Positive" ...
 $ Age     : int  25 17 23 25 24 17 25 20 20 18 ...
> |
```

Step-2: Pre-processing of data using some in-built library in R to prepare the dataset for analysis.



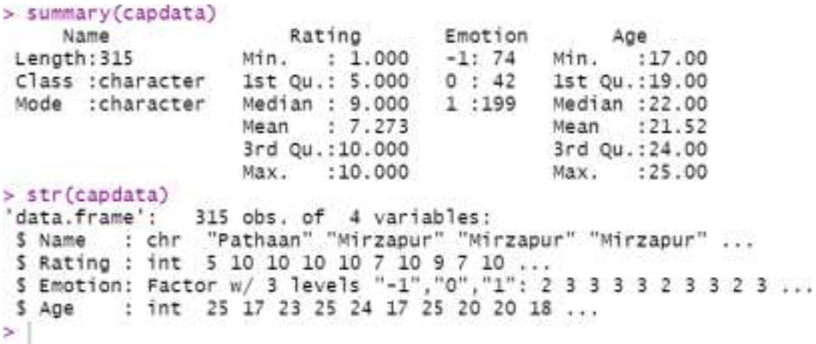
```
> library(plyr)
Warning message:
package ‘plyr’ was built under R version 4.1.3
> capdata$Emotion <- revalue(capdata$Emotion,c("Negative"=-1))
> capdata$Emotion <- revalue(capdata$Emotion,c("Neutral"=0))
> capdata$Emotion <- revalue(capdata$Emotion,c("Positive"=1))
> capdata$Emotion <- as.factor(capdata$Emotion)
> |
```

Step-3: Splitting of dataset in train and test using sample.split() function present inside caTools library.



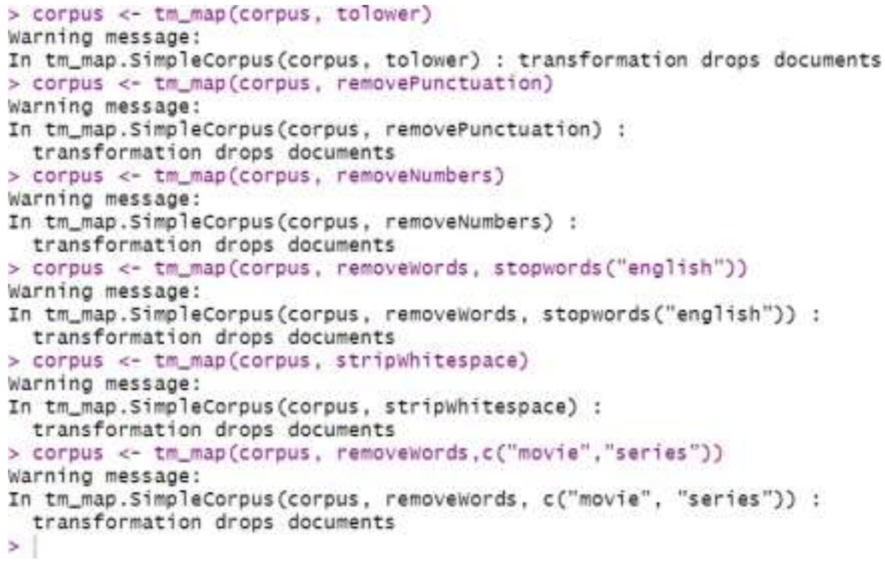
```
> #splitting dataset
> set.seed(123)
> capdata <- capdata[,-2]
> library(caTools)
Warning message:
package 'caTools' was built under R version 4.1.3
> split <- sample.split(capdata, SplitRatio = 0.7)
> train_cl <- subset(capdata, split == "TRUE")
> test_cl <- subset(capdata, split == "FALSE")
> |
```

Step-4: Summary and structure of processed dataset using summary() and str() function.



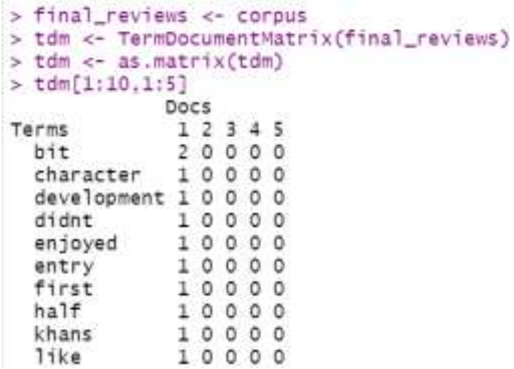
```
> summary(capdata)
     Name               Rating         Emotion        Age       
 Length:315         Min.   : 1.000   -1: 74   Min.   :17.00  
 Class :character   1st Qu.: 5.000   0 : 42   1st Qu.:19.00  
 Mode  :character   Median : 9.000   1 :199   Median :22.00  
                    Mean   : 7.273            Mean   :21.52  
                    3rd Qu.:10.000            3rd Qu.:24.00  
                    Max.   :10.000            Max.   :25.00  
> str(capdata)
'data.frame':   315 obs. of  4 variables:
 $ Name   : chr  "Pathaan" "Mirzapur" "Mirzapur" "Mirzapur" ...
 $ Rating : int  5 10 10 10 10 7 10 9 7 10 ...
 $ Emotion: Factor w/ 3 levels "-1","0","1": 2 3 3 3 3 2 3 3 2 3 ...
 $ Age    : int  25 17 23 25 24 17 25 20 20 18 ...
> |
```

Step-5: Data cleaning on the movie review attribute for the analysis and displaying the word cloud.



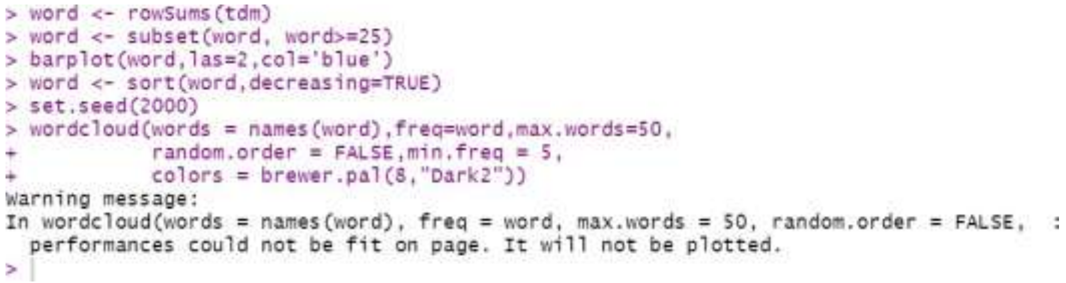
```
> corpus <- tm_map(corpus, tolower)
Warning message:
In tm_map.SimpleCorpus(corpus, tolower) : transformation drops documents
> corpus <- tm_map(corpus, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(corpus, removePunctuation) :
  transformation drops documents
> corpus <- tm_map(corpus, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(corpus, removeNumbers) :
  transformation drops documents
> corpus <- tm_map(corpus, removeWords, stopwords("english"))
Warning message:
In tm_map.SimpleCorpus(corpus, removeWords, stopwords("english")) :
  transformation drops documents
> corpus <- tm_map(corpus, stripWhitespace)
Warning message:
In tm_map.SimpleCorpus(corpus, stripWhitespace) :
  transformation drops documents
> corpus <- tm_map(corpus, removeWords,c("movie","series"))
Warning message:
In tm_map.SimpleCorpus(corpus, removeWords, c("movie", "series")) :
  transformation drops documents
> |
```

Step-6: Converting the reviews into document term for the word cloud using TermDocumentMatrix()

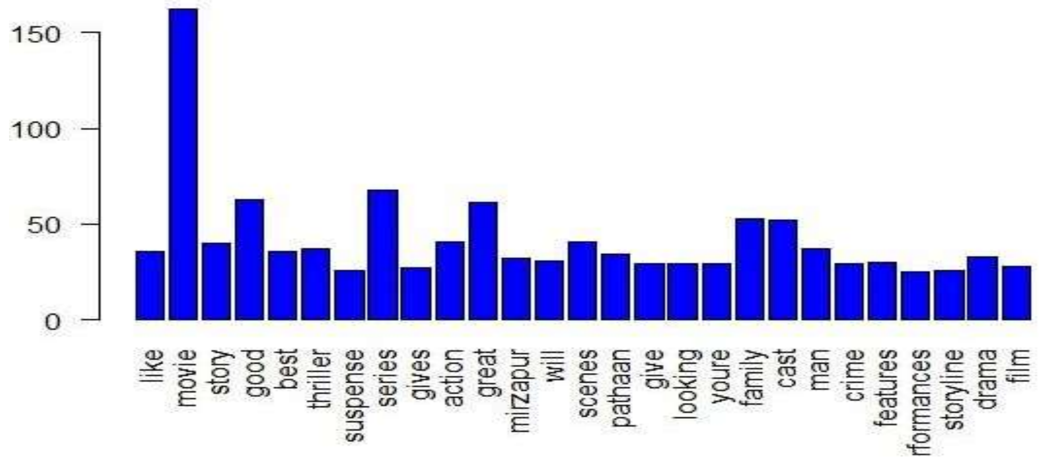


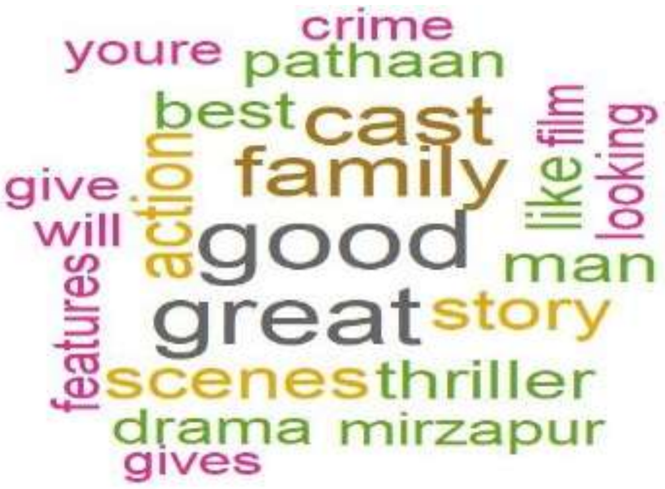
```
> final_reviews <- corpus
> tdm <- TermDocumentMatrix(final_reviews)
> tdm <- as.matrix(tdm)
> tdm[1:10,1:5]
             Docs
Terms         1 2 3 4 5
  bit         2 0 0 0 0
  character   1 0 0 0 0
  development 1 0 0 0 0
  didnt       1 0 0 0 0
  enjoyed     1 0 0 0 0
  entry       1 0 0 0 0
  first       1 0 0 0 0
  half        1 0 0 0 0
  khans       1 0 0 0 0
  like        1 0 0 0 0
```

Step-7: Displaying word bar plot and word cloud

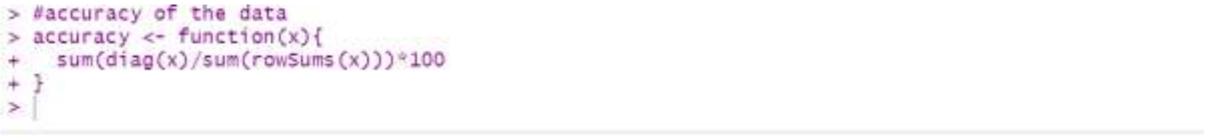


```
> word <- rowSums(tdm)
> word <- subset(word, word>=25)
> barplot(word,las=2,col='blue')
> word <- sort(word,decreasing=TRUE)
> set.seed(2000)
> wordcloud(words = names(word),freq=word,max.words=50,
+           random.order = FALSE,min.freq = 5,
+           colors = brewer.pal(8,"Dark2"))
Warning message:
In wordcloud(words = names(word), freq = word, max.words = 50, random.order = FALSE,  :
  performances could not be fit on page. It will not be plotted.
> 
```



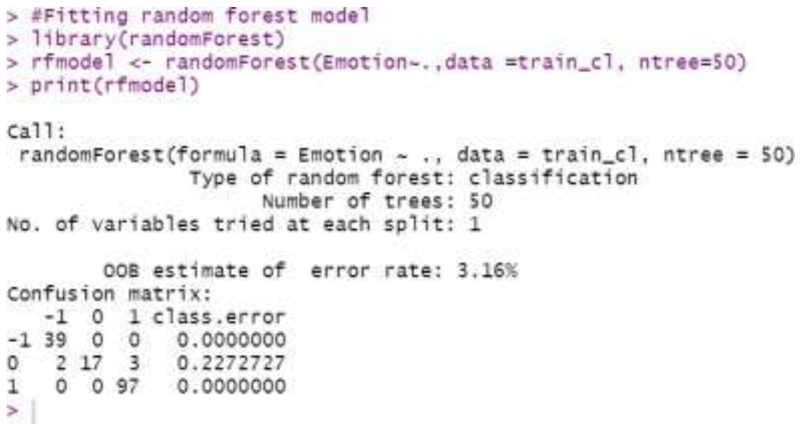


Step-8: Defining the user defined function to calculate the accuracy.



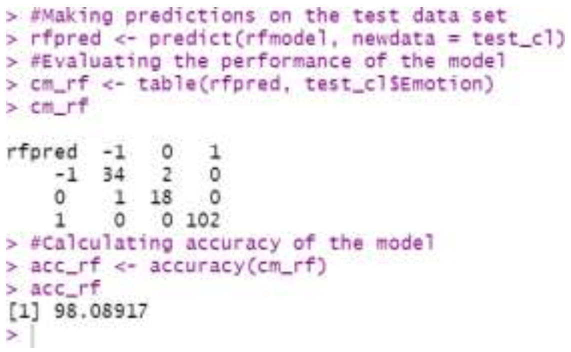
```
> #accuracy of the data
> accuracy <- function(x){
+   sum(diag(x)/sum(rowSums(x)))*100
+ }
> 
```

Step-9: Fitting of Random Forest model using randomForest() function which is defined in randomForest library and if we print the model then it gives the output as confusion matrix.



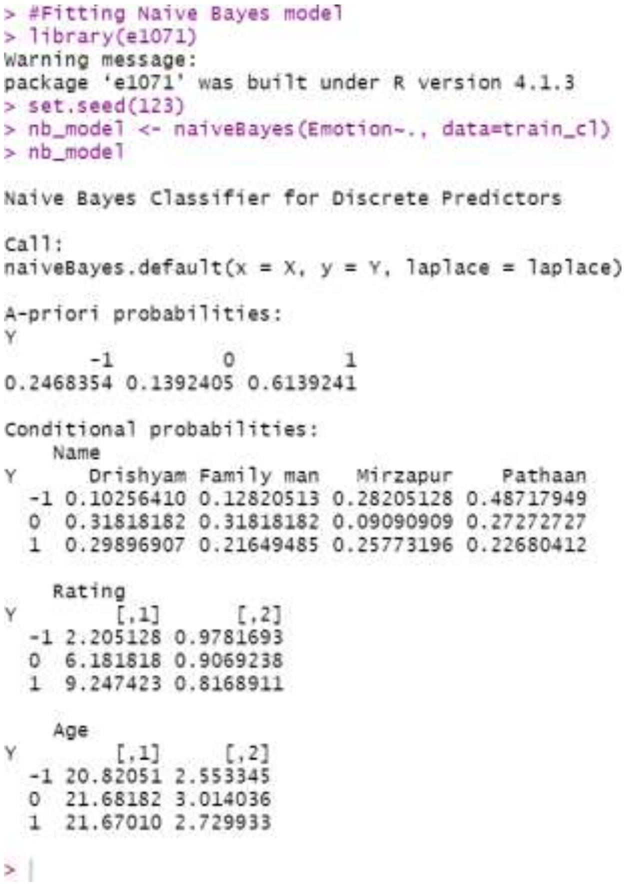
```
> #Fitting random forest model
> library(randomForest)
> rfmodel <- randomForest(Emotion~.,data =train_cl, ntree=50)
> print(rfmodel)

Call:
 randomForest(formula = Emotion ~ ., data = train_cl, ntree = 50)
               Type of random forest: classification
                     Number of trees: 50
No. of variables tried at each split: 1

        OOB estimate of  error rate: 3.16%
Confusion matrix:
   -1  0  1 class.error
-1 39  0  0   0.0000000
0   2 17  3   0.2272727
1   0  0 97   0.0000000
> |
```

Step-10: Making predictions using predict() function and evaluating the performance of model using table() function to display the confusion and further calculating the accuracy using user defined function.



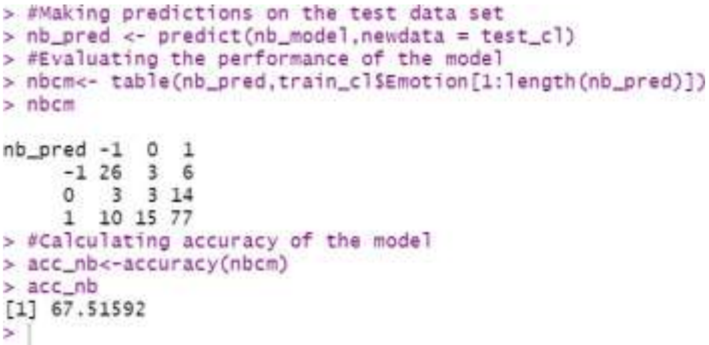
```
> #Making predictions on the test data set
> rfpred <- predict(rfmodel, newdata = test_cl)
> #Evaluating the performance of the model
> cm_rf <- table(rfpred, test_cl$Emotion)
> cm_rf

rfpred  -1   0   1
    -1  34   2   0
    0    1  18   0
    1    0   0 102
> #Calculating accuracy of the model
> acc_rf <- accuracy(cm_rf)
> acc_rf
[1] 98.08917
> |
```

Step-11: Fitting Naive Bayes model using naiveBayes() function which is defined in e1071 library and if we print the model then it gives the A-priori probabilities.



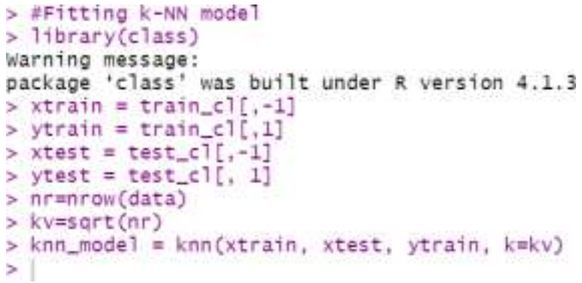
```
> #Fitting Naive Bayes model
> library(e1071)
Warning message:
package ‘e1071’ was built under R version 4.1.3
> set.seed(123)
> nb_model <- naiveBayes(Emotion~., data=train_cl)
> nb_model

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
       -1         0         1 
0.2468354 0.1392405 0.6139241 

Conditional probabilities:
    Name
Y      Drishyam Family man   Mirzapur    Pathaan
  -1 0.10256410 0.12820513 0.28205128 0.48717949
  0  0.31818182 0.31818182 0.09090909 0.27272727
  1  0.29896907 0.21649485 0.25773196 0.22680412

    Rating
Y        [,1]      [,2]
  -1 2.205128 0.9781693
  0  6.181818 0.9069238
  1  9.247423 0.8168911

    Age
Y        [,1]     [,2]
  -1 20.82051 2.553345
  0  21.68182 3.014036
  1  21.67010 2.729933

> |
```

Step-12: Making predictions using predict() function and evaluating the performance of model using table() function to display the confusion matrix and further calculating the accuracy using user defined function.



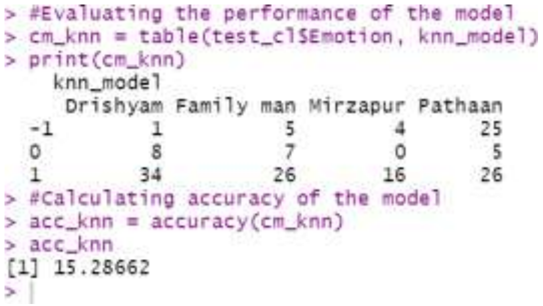
```
> #Making predictions on the test data set
> nb_pred <- predict(nb_model,newdata = test_cl)
> #Evaluating the performance of the model
> nbcm<- table(nb_pred,train_cl$Emotion[1:length(nb_pred)])
> nbcm

nb_pred -1  0  1
     -1 26  3  6
     0   3  3 14
     1  10 15 77
> #Calculating accuracy of the model
> acc_nb<-accuracy(nbcm)
> acc_nb
[1] 67.51592
> |
```

Step-13: Fitting the k-NN model using knn() function defined inside class library and the value of k is defined as the square root of the total number of records in dataset.



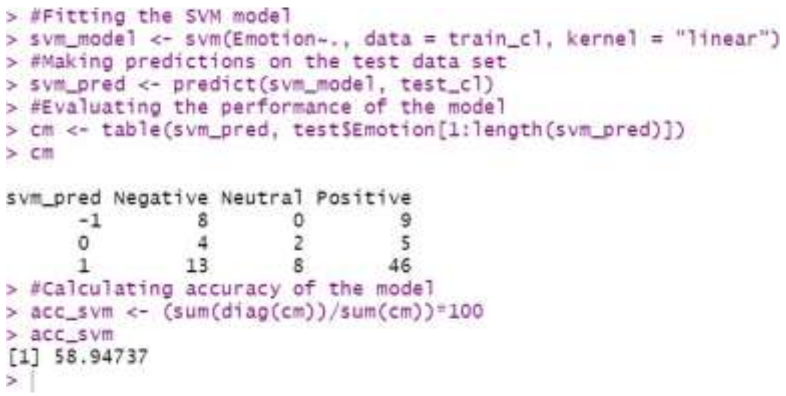
```
> #Fitting k-NN model
> library(class)
Warning message:
package 'class' was built under R version 4.1.3
> xtrain = train_cl[,-1]
> ytrain = train_cl[,1]
> xtest = test_cl[,-1]
> ytest = test_cl[, 1]
> nr=nrow(data)
> kv=sqrt(nr)
> knn_model = knn(xtrain, xtest, ytrain, k=kv)
> |
```

Step-14: Making predictions using predict() function and evaluating the performance of model using table() function to display the confusion matrix and further calculating the accuracy using user defined function.



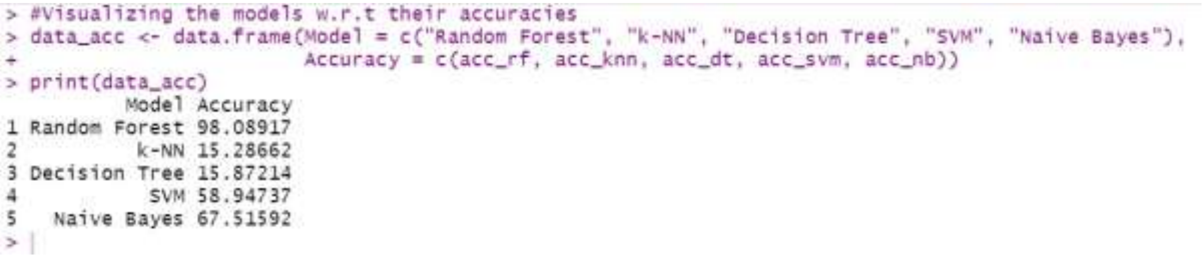
```
> #Evaluating the performance of the model
> cm_knn = table(test_cl$Emotion, knn_model)
> print(cm_knn)
    knn_model
     Drishyam Family man Mirzapur Pathaan
  -1        1          5        4      25
  0         8          7        0       5
  1        34         26       16      26
> #Calculating accuracy of the model
> acc_knn = accuracy(cm_knn)
> acc_knn
[1] 15.28662
> |
```

Step-15: Fitting the SVM model using svm() function defined inside e1071 library, making predictions on test data using predict() function, evaluating the performance of the model using table() function to print the confusion matrix and further calculating the accuracy of the model.

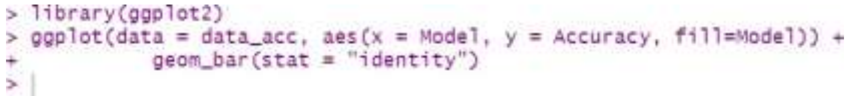


```
> #Fitting the SVM model
> svm_model <- svm(Emotion~., data = train_cl, kernel = "linear")
> #Making predictions on the test data set
> svm_pred <- predict(svm_model, test_cl)
> #Evaluating the performance of the model
> cm <- table(svm_pred, test$Emotion[1:length(svm_pred)])
> cm

svm_pred Negative Neutral Positive
      -1        8       0        9
      0         4       2        5
      1        13       8       46
> #Calculating accuracy of the model
> acc_svm <- (sum(diag(cm))/sum(cm))*100
> acc_svm
[1] 58.94737
> |
```

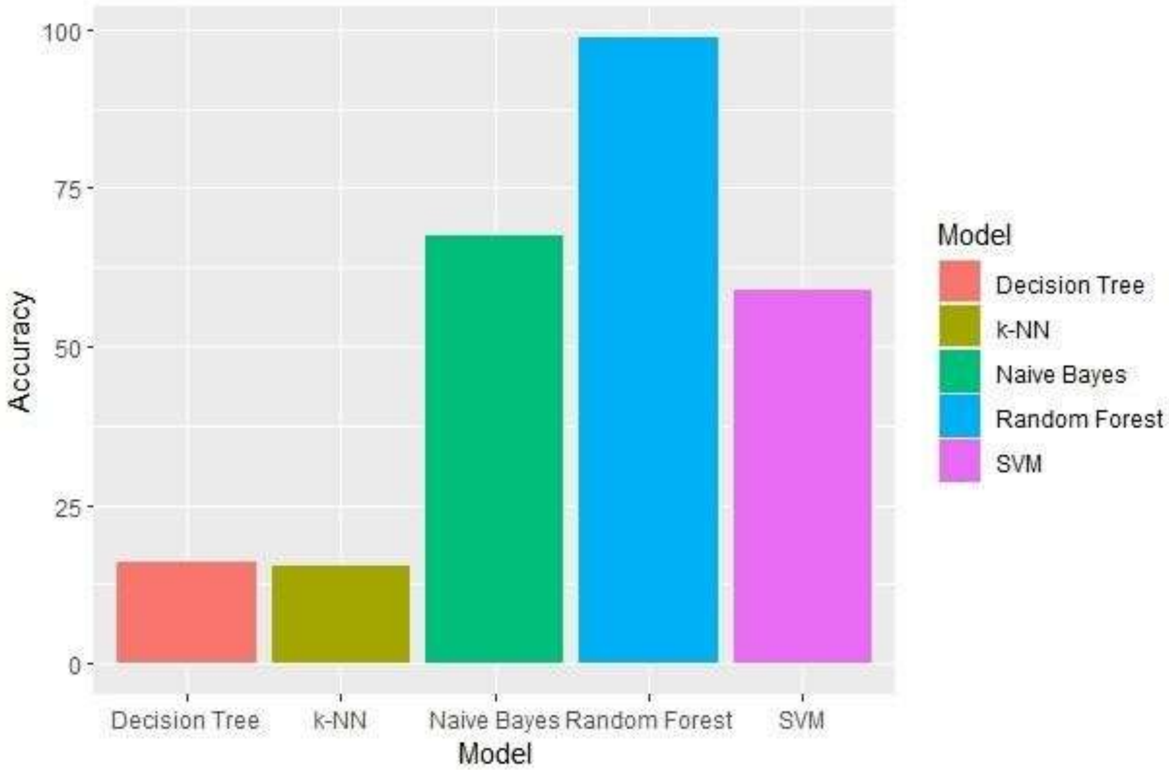
Step-16 : Visualizing the accuracies of the models



```
> #Visualizing the models w.r.t their accuracies
> data_acc <- data.frame(Model = c("Random Forest", "k-NN", "Decision Tree", "SVM", "Naive Bayes"),
+                        Accuracy = c(acc_rf, acc_knn, acc_dt, acc_svm, acc_nb))
> print(data_acc)
          Model Accuracy
1 Random Forest 98.08917
2          k-NN 15.28662
3 Decision Tree 15.87214
4           SVM 58.94737
5   Naive Bayes 67.51592
> |
```



```
> library(ggplot2)
> ggplot(data = data_acc, aes(x = Model, y = Accuracy, fill=Model)) +
+          geom_bar(stat = "identity")
> |
```



REFERENCES

[1] Aishwarya, Parth Wadhwa ,Parbhishek singh,A new sentiment analysis based application for Analyzing reviews of web series and movies.2020 IEEE

[2] Kamal A., 2015, Review Mining for Feature Based Opinion Summarization and Visualization.

[3] Kamal, A. 2013 Subjectivity Classification using Machine Learning Techniques for Mining Feature- Opinion Pairs from Web Opinion Sources. International Journal of Computer Science Issues 10(5), 191- 200.

[4] Basiri, M.E.; Naghsh-Nilchi, A.R.; Ghassem-Aghaee, N. A framework for sentiment analysis in persian. Open Trans. Inf. Process. 2014, 1, 1–14.

[5] Alimardani, S.; Aghaie, A. Opinion Mining in Persian Language Using Supervised Algorithms. J. Inf. Syst. Telecommun. (JIST) 2015.

[6] Amir Hossein Yazdavar, MonirehEbrahimi, Naomie Salim, 2016, Fuzzy Based Implicit Sentiment Analysis on Quantitative Sentences, Faculty of Computing, UniversitiTechnologi Malaysia, Johor, Malaysia, Journal of Soft Computing and Decision Support Systems vol 3:4, pp.7-18.

[7] Palak Baid, Neelam Chaplot, Sentimental Analysis of Movie Reviews using Machine Learning Techniques, ResearchGate, Jaipur, India, December 2017. [8] Palak Baid, Neelam Chaplot, Sentimental Analysis of Movie Reviews using Machine Learning Techniques, ResearchGate, Jaipur, India, December 2017.

[9] A. Hogenboom, F. Frasincar, F. de Jong, and U. Kaymak. 2015, Using Rhetorical Structure in Sentiment Analysis, Communications of the ACM, vol. 58, no. 7, pp. 69–77.

[10] Palak Baid, Neelam Chaplot, Sentimental Analysis of Movie Reviews using Machine Learning Techniques, ResearchGate, Jaipur, India, December 2017