**Public Sentiment Analysis on IMDB Reviews Using Natural Language Processing**

**A Project Report**

***Submitted by:***

**Praviveek Ray (2041018084)**

***in partial fulfillment of the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



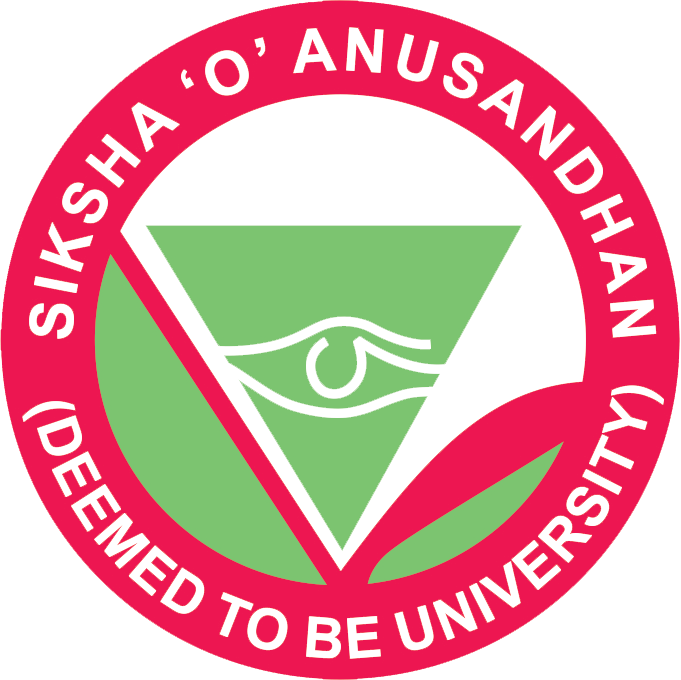
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Faculty of Engineering and Technology, Institute of Technical Education and Research**

**SIKSHA ‘O’ ANUSANDHAN (DEEMED TO BE) UNIVERSITY**

**Bhubaneswar, Odisha, India**

**(June 2024)**



**CERTIFICATE**

This is to certify that the project report titled **“Public Sentiment Analysis on IMDb Reviews Using Natural Language Processing”** being submitted by **Mihika Satpathy (2041019174),** **Samridhi Pattanayak (2041016193), Praviveek Ray (2041018084), Kamrup Mohapatra (2041013222) of sec ‘F’** to the Institute of Technical Education and Research, Siksha ‘O’ Anusandhan (Deemed to be) University, Bhubaneswar for the partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering is a record of original confide work carried out by them under my/our supervision and guidance. The project work, in my/our opinion, has reached the requisite standard fulfilling the requirements for the degree of Bachelor of Technology.

The results contained in this project work have not been submitted in part or full to any other University or Institute for the award of any degree or diploma.

(Name and signature of the Project Supervisor)

Department of Computer Science and Engineering

Faculty of Engineering and Technology;

Institute of Technical Education and Research;

Siksha ‘O’ Anusandhan (Deemed to be) University

**ACKNOWLEDGEMENT**

We express our heartfelt gratitude to our project supervisor, Dr. Ashutosh Tripathi, for his exceptional guidance and his exceptional guidance and steadfast support. His expertise and patience were instrumental in shaping the direction and outcomes of our final-year group project. Their encouragement and willingness to assist in procuring the necessary materials were crucial to overcoming the challenges we faced during this research. We would like to acknowledge the faculty member for their exceptional guidance as well as their, insightful feedback, and our fellow group members for teamwork, dedication, and the enriching discussions that have broadened our understanding and approach to the project. It is through their collective efforts that our project has reached its outcome.



**Place: Bhubaneswar Signature of Student**

**Date: 20/06/2024**

**DECLARATION**

We declare that this written submission represents our ideas in our own words and where other’s ideas or words have been included, we have adequately cited and referenced the sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented fabricated or falsified any idea/fact/source in our submission. We understand that any violation of the above will cause disciplinary action by the University and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken when needed.

**Praviveek Ray - 2041018084**

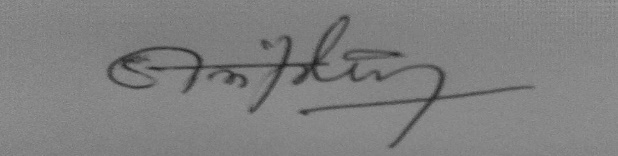
Signature of Student with Registration Numbers

Date: 20/06/2024

**REPORT APPROVAL**

This project report titled **“Public Sentiment Analysis on IMDb Reviews using Natural Language Processing”** being submitted by **Praviveek Ray (2041018084) of sec ‘F’** is approved for the degree of *Bachelor of Technology in Computer Science and Engineering*.

**Project Coordinator**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PREFACE**

This project aims to create a sophisticated sentiment analysis system tailored for IMDb movie reviews. The system uses modern deep learning and machine learning techniques to identify evaluations as either favorable or negative reliably. Sentiment analysis, sometimes called opinion mining, is a crucial process for comprehending user attitudes and preferences because it entails identifying emotions and viewpoints inside text. The system uses several strategies along with various embedding approaches to improve sentiment classification precision.

To determine the most correct strategy, the study investigates deep learning and machine learning models, applying Graph Convolutional Networks (GCNs) as one of the latter. The analysis is further improved by including several embeddings, word2vec, and TF-IDF, producing a very trustworthy tool for sentiment evaluation.

By rigorously evaluating and optimizing the model, this research intends to create a standard for sentiment analysis in movie reviews and enhance natural language processing technology. The study highlights the value of integrating state-of-the-art and conventional techniques to raise the accuracy and stability of sentiment analysis systems.

Apart from its technological advancements, the initiative provides an invaluable tool for understanding and forecasting the reactions of moviegoers. The findings demonstrate how combining several machine learning and deep learning approaches can improve sentiment analysis's Precision[15] and resilience.

All things considered, this research is a major advancement in the field of complex sentiment analysis algorithms, offering a trustworthy and perceptive tool for examining IMDb movie reviews.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | Title Page | i |
|  | Certificate | ii |
|  | Acknowledgment | iii |
|  | Declaration | iv |
|  | Report Approval | v |
|  | Preface | vi |
|  | Individual Contributions | vii |
|  | Table of Contents | viii |
|  | List of Figures | ix |
|  | List of Tables (optional) | x |
| **1.** | **INTRODUCTION** | 1 |
|  | * 1. Introduction   2. Project Overview   3. Motivation(s)   4. Uniqueness of the Work   5. Report Layout | 1  1  2  3  4 |
| **2.** | **LITERATURE SURVEY** | 5 |
|  | 2.1 Existing System  2.2 Problem Identification | 5  6 |
| **3.** | **MATERIALS AND METHODS** | **8** |
|  | 3.1 Dataset(s) Description  3.2 Schematic Layout  3.3 Methods  3.4 Tools/Technologies  3.5 Evaluation Measures | 8  9  17  21  23 |
| **4.** | **EXPERIMENTATION AND RESULTS/APPLICATION/SYSTEM DESIGN AND OUTPUTS** | 26 |
|  | 4.1 System Specification  4.2 Parameters Used (if any)  4.3 Results and Outcomes  4.4 Result Analysis and Validation | 26  26  27  29 |
| **5.** | **CONCLUSIONS** | 32 |
| **6.** | **REFERENCES** | 33 |
| **7.** | **APPENDICES** | 35 |
| **8.** | **REFLECTION OF THE TEAM MEMBERS ON THE PROJECT** | 54 |
| **9.** | **SIMILARITY REPORT** | 56 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| NO | FIGURE NAME | | PAGE NO |
| 1 | Complete flow diagram of our model step-by-step | | 9 |
| 2 | Logistic Regression Graphical Representation | | 18 |
| 3 | Schematic layout of random Forest | | 19 |
| 4 | Artificial Neural network layout | | 19 |
| 5 | Long-Short-Term Memory Layout | | 20 |
| 6 | Graph Convolutional Network Layout | | 21 |
| 7 | Confusion Matrix | | 24 |
| 8 | Visualization using a Heatmap of the model | | 28 |
| 9 | Performance Comparison of the model | | 31 |
|  | |
|  | |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| NO | TABLE NAME | PAGE NO |
| 1 | Top found data | 10 |
| 2 | System Specification | 26 |
| 3 | F1-Score, Recall, Precision of all the models in tabular form (Weighted Average of each) | 28 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **INTRODUCTION**

**1.1 Introduction**

Millions of people worldwide enjoy movies as entertainment, and the movie industry contributes to the global economy. Movie reviews have become crucial in shaping public opinion and influencing box office success with the rise of social media and online platforms. The popularity and profitability of a movie, as well as the careers of actors and filmmakers involved, can be influenced by the sentiment expressed in these reviews. It takes a lot of time and labor to analyze and process movie reviews. Movie reviews often contain complex language, nuanced sentiment, and differing levels of subjectivity. It is difficult for filmmakers and industry professionals to keep track of public opinion due to the sheer volume of reviews generated by the public. There is a growing need for automated sentiment analysis [1] tools that can analyse movie reviews and determine their sentiment. This can help filmmakers understand public opinion, identify trends, and make data-driven decisions about their projects.

Machine learning is a promising approach to sentiment analysis [1] as it can learn patterns and relationships in large datasets and make predictions based on these insights. Graph convolutional networks [2] have shown great promise in sentiment analysis as they can effectively capture the complex relationships between words and phrases in text data. The project's primary objective is to develop a machine-learning model that can accurately analyze movie reviews and determine their sentiment using GCNs.

* 1. **Project Overview**

The goal of our study is to apply cutting-edge methods and models to improve sentiment analysis for IMDb movie reviews. Our goal is to get over existing obstacles, like the challenge of categorizing neutral writings and the fluidity of language on social networking platforms. We improve the resilience and accuracy of sentiment classification by utilizing sophisticated model Graph Convolutional Networks (GCNs) in conjunction with hybrid Feature Extraction [13] techniques used such as TF-IDF and word2vec embeddings. A thorough sentiment analysis system that can accurately classify evaluations as positive, negative, or neutral will be developed with the use of a comparative analysis of machine learning and Deep learning [9]models. This analysis will shed light on the efficacy of these models. The goal of this project is to make a substantial contribution to the field of natural language processing by providing useful applications in market analysis, user sentiment analysis, and content recommendations.

• There is a Data Collection[12] of movie reviews from various sources.

• The data can be pre-processed by tokenizing the text, removing stop words, and converting all text to lowercase.

• The GCN-based sentiment analysis model can be implemented using the PyTorch library.

• Model Evaluation uses metrics accuracy, Precision[14], recall, and F1-Score[16]to evaluate the model's performance.

• The GCN model is compared with other machine learning and Deep learning [9]models Artificial neural networks, Recurrent neural networks, Logistic Regression, and Random Forest.

* 1. **Motivations**

Improving sentiment analysis for IMDb movie reviews to obtain deeper insights into audience preferences and emotions is the primary challenge that motivates our effort. Filmmakers, marketers, and other industry players need to have this knowledge to make wise choices regarding the production of content, advertising tactics, and audience involvement. Our sophisticated sentiment analysis algorithm aims to overcome the present shortcomings in correctly categorizing evaluations as neutral, negative, or positive.

We used cutting-edge methods from machine learning and natural language processing (NLP) to accomplish this. We used hybrid Feature Extraction [13] techniques word2vec [3] embeddings and TF-IDF. These techniques are crucial for capturing contextual nuances as well as semantic meanings in movie reviews, where minute changes in emotion can have a big impact on how the audience reacts.

A noteworthy breakthrough in our Graph Convolutional Networks (GCNs) is one method. These networks allow us to represent complicated dependencies that standard models would overlook, allowing us to model complex interactions between words or concepts within reviews. This feature is essential for improving sentiment classification's Precision[14] and dependability, especially when feelings are complex or unclear.

Our initiative seeks to provide stakeholders with practical insights into audience sentiments by increasing the accuracy of sentiment analysis. Consequently, this improves decision-making procedures concerning marketing campaigns, audience engagement tactics, and content creation. In the end, our work advances natural language processing (NLP) technology and its useful applications in marketing, consumer behavior analysis, and entertainment. Employing meticulous testing and assessment, we aim to provide a novel benchmark for sentiment analysis platforms customized particularly for IMDb movie reviews, showcasing the revolutionary potential of cutting-edge NLP methods in comprehending and forecasting viewer reactions.

* 1. **Uniqueness of the Work**

Our project stands out in several key ways, primarily through its innovative approach to sentiment analysis of IMDb movie reviews using advanced natural language processing (NLP) techniques. Unlike traditional sentiment analysis systems that often rely on basic classifiers and simplistic Feature Extraction [13] methods, our work integrates cutting-edge methodologies to achieve superior accuracy and depth of analysis.

One of the unique aspects of our project is the incorporation of Graph Convolutional Networks (GCNs) alongside traditional machine learning models. GCNs enable us to model and capture intricate relationships and dependencies between words or concepts within movie reviews, leveraging the inherent structure of language data more effectively than traditional methods. This capability not only enhances the accuracy of sentiment classification but also provides deeper insights into the contextual nuances of audience sentiments.

Furthermore, our project emphasizes the use of hybrid Feature Extraction [13] techniques used in this study including TF-IDF and word2vec [3] embeddings. These methods enable us to encode semantic meanings and capture nuanced expressions within the text, which is critical for understanding the varying degrees of positivity or negativity in movie reviews. By combining these advanced Feature Extraction [13] techniques with GCNs, we achieve a more robust and reliable sentiment analysis system that outperforms conventional approaches.

Additionally, our project contributes to the field by conducting a comprehensive comparative analysis of various machine learning and Deep learning [9]models specifically tailored for IMDb movie reviews. This includes evaluating model performance metrics accuracy, Precision[14], recall, and F1-Score[16], and providing a thorough assessment of each model's effectiveness in real-world applications.

Overall, our project's uniqueness lies in its integration of state-of-the-art NLP techniques, including GCNs and hybrid Feature Extraction [13] methods, to advance the accuracy, depth, and applicability of sentiment analysis in the domain of movie reviews. By pushing the boundaries of existing methodologies, we aim to set a new benchmark for sentiment analysis systems, offering valuable insights for filmmakers, marketers, and industry stakeholders to better understand audience sentiments and enhance decision-making processes.

* 1. **Report Layout**

Section I introduced a sophisticated sentiment analysis system designed for evaluating IMDb movie reviews using advanced Deep learning [9]and machine learning techniques. It outlined the significance of sentiment analysis in enhancing audience engagement, refining content recommendations, and optimizing marketing strategies in modern data-driven contexts.

Section II, the Literature Review, provided a comprehensive survey of recent advancements in sentiment analysis and natural language processing (NLP). It synthesized key findings from existing research, discussing various sentiment classification methods, text embedding techniques used in this study Word2Vec and TF-IDF, and evaluating the efficacy of machine learning[4] and Deep learning [9]models.

Section III detailed the methodology employed in developing the sentiment analysis system. It described the system architecture, integration of models including Graph Convolutional Networks (GCNs), data preprocessing steps, feature engineering techniques, and criteria for selecting models.

Section IV rigorously evaluated the sentiment analysis system's performance. It presented results from comparative experiments across multiple models and embedding methods, using metrics like accuracy, Precision[14], recall, and F1-Score[16]. Visualization used in this study was confusion matrices that illustrated the models' performance.

Section V interpreted experimental findings and discussed their implications for filmmakers, marketers, and industry stakeholders. It explored how the sentiment analysis system could inform decision-making, enhance audience targeting, and influence content creation and distribution strategies.

Section VI, the Conclusion, summarizes the achievements and contributions of the sentiment analysis system. It emphasized its capability to extract valuable insights from IMDb movie reviews using advanced computational techniques and outlined future research directions and potential innovations in sentiment analysis.

This structured layout guided readers through the development, evaluation, and implications of the sentiment analysis system for IMDb movie reviews, highlighting its innovative contributions and potential impact on industry practices.

1. **LITERATURE SURVEY**

**2.1 Existing System**

The original work on the dataset involved the creation of word vectors for the polarity classification of reviews and the clustering of semantically comparable terms using unsupervised learning. This method was also used to predict reviewer scores, identify review polarity, and categorize reviews into many groups. Neutral text classification proved to be difficult when employing Random Forest and SVM classifiers. Context-aware models, hierarchical architectures, transfer learning, and performance-enhancing ensemble techniques are further improvements.

In our research, we expanded our sentiment analysis to classify text into five distinct human emotions using Word2Vec embeddings. Despite our efforts, the classification Accuracy[6]for emotions was notably low. This underscores the difficulty in effectively capturing complex emotional nuances solely through textual analysis methods like Word2Vec. Further exploration of advanced techniques may be necessary to improve emotion classification Accuracy[6]in sentiment analysis applications.

Hassan and Mahmood (2017) [1] researched the application of Deep learning [9]to short-text sentiment analysis. They achieved notable gains in sentiment classification accuracy by applying a deep learning approach and using a dataset from the Third International Conference on Control, Automation and Robotics (ICCAR). Their research on several model architectures and preprocessing methods to improve sentiment analysis system performance was presented in Nagoya, Japan.

In 2017, A. Kiritchenko and S. M. Mohammad [2] looked at sentiment analysis's gender bias. Their study examined the existence and consequences of gender bias in sentiment analysis algorithms, and it was published on ResearchGate in November 2017. They discovered substantial gender biases that affect the fairness and accuracy of sentiment predictions by carefully examining sentiment-labeled datasets and sentiment analysis technologies. The study made clear that to reduce these biases and enhance sentiment analysis systems' overall performance, more inclusive and representative training data are required.

A. Narayanan et al. (2019) [3] carried out an extensive investigation on sentiment analysis, examining a range of techniques and resources employed in the discipline. The study examined a variety of methods for assessing sentiment in textual data, including Lexicon-based approaches[8]es, hybrid strategies, and machine-learning

techniques. It was published on ResearchGate in April 2019. The authors also examined several frequently used sentiment analysis tools, assessing their suitability and efficacy in various settings. Their work sheds insight into

the difficulties and developments in the field of sentiment analysis research and advances our understanding of how it might be used in a variety of contexts.

In a survey on COVID-19 contact-tracking apps, N. Ahmed et al. (2020) [4] looked at the apps' technological foundations, privacy concerns, and efficacy. The study examined Bluetooth and GPS-enabled worldwide apps, underlining the trade-offs between privacy and public health and making suggestions for future developments.

Kumar et al. (2020) [5] investigated the impact of age and gender on sentiment analysis using machine learning techniques. Their study, conducted on a diverse dataset, employed various machine learning models to analyze how sentiment varies across different age groups and genders. The results demonstrated significant variations in

sentiment classification accuracy based on these demographic factors, highlighting the importance of considering age and gender in sentiment analysis models to enhance their performance and Reliability[6].

Buchan, N. A., Richardson, N. P., and Gorsuch, R. M. (2023) [6] investigated the determination of critical thresholds in social networks. Utilizing data published in the Proceedings of the National Academy of Sciences of the United States of America (vol. 120, no. 23, pp. e10280647), they identified key factors influencing the stability and dynamics of these networks. Their study highlighted the importance of understanding these thresholds for predicting network behavior and implementing effective interventions.

Overall, the existing research suggests that sentiment analysis can provide valuable insights into stock market trends. However, there is still a need for further research to refine and improve the accuracy of sentiment analysis techniques for predicting stock prices.

* 1. **Problem Identification**

Current sentiment analysis systems, as discussed in several research works, often utilize classifiers like Random Forest and Support Vector Machines (SVMs) for the classification of reviews. These systems rely on various Feature Extraction [13] techniques, including TF-IDF and word embeddings, to convert textual data into numerical representations. A significant limitation noted in these studies is the exclusion of a neutral category in classification tasks. Neutral texts, which are assumed to lie close to the boundary of binary classifiers, pose a unique challenge as they are disproportionately hard to classify accurately. This oversight can lead to a reduction in the overall accuracy and robustness of sentiment analysis models.

The increasing availability of sentiment analysis tools, both free and commercial, has facilitated their widespread adoption. With the rise of microblogging platforms, sentiment analysis is extensively used to gauge public sentiments and draw valuable inferences. One notable application is the analysis of Twitter data to understand political sentiment, such as during the German Federal elections. These tools have proven instrumental in capturing the public's mood and predicting trends based on social media activity.

However, despite the advancements, there are still significant challenges that need to be addressed. For instance, existing systems often struggle with context and sarcasm, which can lead to misclassification. Moreover, the rapid evolution of language and slang on social media platforms necessitates continuous updates to the sentiment analysis algorithms.

In our project, we aim to overcome some of these challenges by integrating diverse embeddings such as word2vec and TF-IDF and exploring advanced models like Graph Convolutional Networks (GCNs). This approach not only enhances the Precision[14] of classifying IMDb movie reviews but also provides deeper insights into the underlying sentiments. By comparing machine learning and deep learning models, our research highlights the potential for improved sentiment analysis through the integration of cutting-edge and traditional techniques, setting a new benchmark for future developments in this field.

1. **MATERIALS AND METHODS**

**3.1 Dataset Description**

The dataset used in this research is an extensive set of movie reviews that were obtained from IMDb, the well-known website that provides information on movies and television shows. IMDb is an excellent resource for sentiment analysis in the context of movie evaluations since it offers a sizable collection of user-generated reviews covering a broad range of genres, languages, and release years. To obtain structured and pertinent textual data while closely following ethical standards and IMDb's terms of service, a methodical web scraping procedure was used to obtain the dataset.

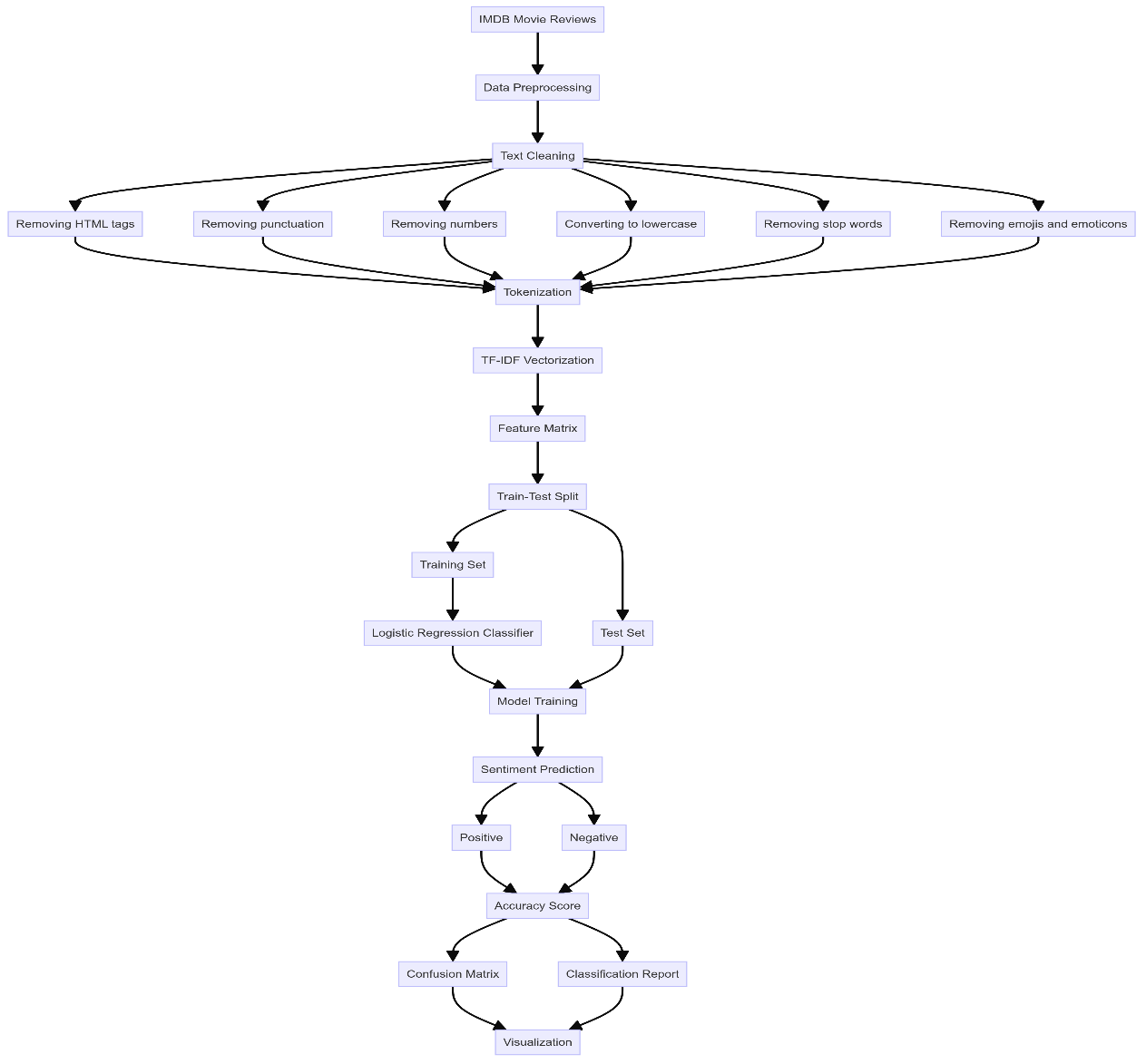
Beautiful Soup[11] was used in conjunction with other Python[7] modules to enable the extraction of review texts and related metadata, including movie titles, user ratings, review dates, and sometimes user demographics, during the web scraping process. This strategy made sure the dataset was large and organized, allowing for reliable modeling and analysis. The dataset used in this study consists of 54,565 rows and 2 columns, sourced from IMDb, a widely recognized platform for movie reviews and ratings. The dataset is formatted in CSV, a standard format for organizing tabular data, facilitating easy accessibility and manipulation. The two primary columns in the dataset are 'Review' and 'Rating'. The 'Review' column contains the textual content of the movie reviews gathered through web scraping, while the 'Rating' column corresponds to the user-provided ratings associated with each review.

This structured format and comprehensive scope enable robust analysis and modeling in sentiment analysis tasks, leveraging the diverse and extensive nature of IMDb's user-generated content. Ethical considerations and adherence to IMDb's terms of service were paramount throughout the data acquisition process, ensuring responsible use of user-contributed reviews while maintaining privacy and confidentiality standards. This dataset serves as a valuable resource for exploring sentiment trends in movie critiques and advancing natural language processing techniques within the domain of film evaluation.

In general, the IMDb data [10]set that was taken out offers a large and varied supply of movie reviews that may be used to test and train sentiment analysis algorithms. The dataset is a strong basis for researching sentiment trends in movie assessments and enhancing natural language processing in the context of film criticism because of its comprehensive nature, adherence to ethical norms, and quality control methods.

* 1. **Schematic Layout**

The methodical procedure for creating a sentiment analysis model with IMDB movie reviews is shown in the flowchart. First, information is gathered from IMDB, including ratings and reviews, which serve as the foundation for the analysis. Following extensive preparation, the data is cleaned up by removing HTML tags, punctuation, and numerals, changing text to lowercase, getting rid of stop words, and managing emoticons and emojis. After preprocessing, the text is tokenized into discrete words or phrases, and TF-IDF vectorization is used to convert the text into numerical vectors. The feature matrix, which is divided into training and test sets for model creation and evaluation, is made up of these vectors. Because of its capacity to simulate the correlation between the retrieved characteristics and sentiment labels (positive or negative), a logistic regression classifier is selected. The model undergoes training on the training set and is then used to test set sentiment prediction. Evaluation metrics are used to evaluate the model's performance and provide information about how well it classifies the sentiment of IMDB movie reviews. Examples of these metrics are the accuracy score, Confusion Matrix[17], and classification report. This methodical technique guarantees a strong sentiment analysis system that can accurately classify movie reviews according to their sentiment.



*Figure 1. Complete the Flow diagram of our model step-by-step*

**3.2.1 Data Collection**

The primary source of information for this project's Data Collection[12] was IMDB, where a complete dataset of movie reviews was assembled. A wide range of viewpoints on different movie genres, languages, and release dates was offered by IMDB, which is well-known for its sizable collection of user-generated reviews and ratings. Recognized for compiling reviews from critics and viewers, IMDb provided an additional insightful viewpoint on public reaction to films. To ensure reliable model training and assessment, the dataset included a significant number of reviews. To make supervised learning tasks easier, reviews were labeled with sentiments that were positive, negative, and neutral. We used online scraping methods and automated scripts to gather the data, closely following platform terms of service and ethical guidelines to preserve data integrity and protect user privacy. Techniques for data aggregation and anonymization were applied when appropriate to protect private data. With the use of these many sources and methodical data-gathering procedures, we created a complete dataset that enabled machine learning algorithms for reliable sentiment analysis of movie reviews.

*Table 2. Top-found data*

|  |  |
| --- | --- |
| **RATINGS** | **REVIEWS** |
| 10 | I'll admit I raised an eyebrow when I saw that Pattinson was cast, but I eat my words, he was awesome, and hopefully will play the part a few more times.  This film blew me away, exciting, fast paced, surprisingly gritty, and genuinely had an awesome story.  The franchise somehow felt renewed here, even if you're not a fan of superhero movies, you will love this.  Andy Serkis was genuinely awesome, but when isn't he? The acting throughout was fantastic, with plenty of well known faces.  Great music, great special effects, I've already booked to see it again.  10/10. |
| 9 | NaN |
| 7 | NaN |
| 9 | NaN |
| 9 | The Riddler(Paul Dano, spot-on. How did it take this long for him to get a role like this?) targets public officials, revealing their corruption, and killing them in gruesome fashion(how did this get away with a PG-13 again? Oh, right, as long as you don't show the details, then you can get away with almost anything, by now. Even a barely toned down Jigsaw). In order to stop him, it will be necessary for the Bat and the Cat(with amazing chemistry), together, to stop the rat. Thanks, Matt.  I really did not think that we needed yet another film dealing with Bruce Wayne's alter-ego. How about Nightwing? I know it's not likely, but part of me still holds out hope that Grayson will be turned into a feature(if you've never checked out the trailer on YouTube, you're missing out). But somehow, this managed to convince me. Fingers crossed for at least one sequel. Robert Pattinson is incredible here. Seriously, can we just stop freaking out every time the role is recast? The closest we've come to someone who shouldn't be doing it is George Clooney, and it's not like anybody came out of that flick looking good. He took that job after doing From Dusk till Dawn. There was good reason to think that he was going to be badass. Honestly, everyone here gives a strong performance.  Something that will definitely appeal to some more than others, is the genre and tone. This is essentially Se7en meets Zodiac. It is not paced like a typical massive blockbuster. While the action is great, especially the martial arts, not to mention that car chase, there's less of it and it is smaller scale than for example the Nolan trilogy(which also has far more escalation). Essentially, this just isn't focused that much on that aspect of the titular icon. And I think we can all agree that we've had some great entries that deliver that. This is more interested in conveying to the audience that there's a reason he's called the world's greatest detective. We've barely seen it outside of the animated ones before, so I'm very happy with that choice by them.  In addition to what I've already mentioned, this features some drugs, suggestive material and strong language. I recommend this to any fan of Batman. 9/10. |
| 7 | NaN |
| 8 | THE BATMAN (2022) \*\*\* Robert Pattinson, Zoe Kravitz, Jeffrey Wright, Colin Farrell, Paul Dano, John Turturro, Andy Serkis, Peter Sarsgaard. Filmmaker Matt Reeves pulls out all the stops in this even darker adaptation of the Bob Kane comic book hero with Pattinson standing tall and brooding as Bruce Wayne employing vigilante style justice as The Caped Cursader while Gotham quakes under the siege of the homicidal Riddler (Dano affectively disturbing) and gaining an unlikely ally in Selina Kyle (ass-kicking Kravitz in pre-Catwoman mode). Arresting visuals thanks in large part to ace cinematography by Greig Fraser and James Chinlund's production design all aided by the foreboding score by Michael Giacchino full of bombast and dread. Farrell, under a ton of unrecognizable prosthetics, has a field day in getting in touch with his inner Robert De Niro as The Penguin. By all means see it in IMAX!  253 out of 409 found this helpful. Was this review helpful? Sign in to vote.  Permalink |
| 8 | Always been a "Batman" fan as the D. C. legend is my favorite superhero and watching all of the movies from the classic 1989 one, every so often the fans see that the series changes tone and feel as it's rebooted. "The Batman" directed by Matt Reeves is a dark take with conflict as "Batman" deals with a city underworld of secrets, while serial killings are happening, and the hope and chance for romance struts right in front of him.  Gotham has a serial killer on the loose and with each victim found strange puzzle like word clues are left, as "Batman" investigates the mystery and things become complex as it's all tied into a world of city officials and underworld crime figures are connected to them. Plus secrets come up as "Batman" himself is wrapped up into this wicked game.  The film chemistry and cast of characters are top notch with young star Robert Pattinson filling out the bat suit just fine and Zoe Kravitz is raw and outspoken as Selina Kyle a slash friend and foe love interest of the bat. Colin Farrell was well made up as the Penguin thru it all combined with fire blazing action and kick butt punches "The Batman" is a compelling drama with suspense and revealing secrets of Gotham city officials as crime boss Carmine Falcone is a lead figure to all of this.  "The Batman" is in depth and dark a new reboot style take on the D. C legend as it's a good film still not the best, still if a "Batman" and D. C. fan you will enjoy as the film rubs off a feel from many of the graphic novels, as to get more understanding read "The Dark Knight Returns" or "The Long Halloween". |
| 6 | The headline can be taken two ways. If you remove any hype or expectations, saying it's a good movie is positive. But if you are hoping for an amazing new Batman iteration, good means it wasn't great, let alone amazing.  This review will come off as negative but I did enjoy it. I love the tone. It's super dark and grounded in reality, even more than the Nolan trilogy. And it feels true to the character. Zoe Kravitz and Jeffrey Wright are great casting choices. It's a mostly well-made film and a good addition to the franchise. And the Bat suit looks awesome.  At the same time, nothing about this movie wowed me. I did not connect with the characters, especially the villain. The relationships seem undeveloped and not organic. I was not nearly as emotionally invested in the story as I should have been. It mostly feels surface level.  If you're a huge Batman fan, I think you will really enjoy it. But as a movie, I don't think it fires on all cylinders. The Nolan trilogy still reigns supreme. (1 viewing, fan early screening 3/2/2022) |
| 1 | Slow, pointless and not worth watching. Just terrible!  2 out of 7 found this helpful. Was this review helpful? Sign in to vote.  Permalink |
| 1 | I really like going to the cinema and watching movies , so far i regret two times to sit in a cinema and pay for a movie. First one was "Mother!" last year ... now this one , but this one is 2 time worse than the mother ...... Please don't waste your time and money with this piece of \*\*\*\*\* my Girlfriend fell asleep during the movie :D2 out of 6 found this helpful. Was this review helpful? Sign in to vote. Permalink |
| 4 | I seriously dont know how Keanu Reeves starts at thisðŸ˜‘  0 out of 2 found this helpful. Was this review helpful? Sign in to vote.  Permalink |
| 7 | Siberia is film making at it's best. I was quite surprised to see this film garnering negative reviews almost across the board. It seems most likely that people went into this film with John Wick on their minds, which I think is unavoidable and automatic. Even I began watching Siberia with thoughts of Wick in the back of my head. It's hard not to. However this is not John Wick or even an action film. The association with the Wick Series is apparent throughout the film. The same sets, clothing, accents, and even the main character's demeanour lends it's self to comparison. I think the association with Wick has doomed Siberia by creating a failure of expectations.  Had this film been made in distance from the Wick series, it would be easier for viewers to appreciate the nuances that for me, made this film so powerful. On so many levels this film shined, in character development, in story, or how the film maker took a traditional Hollywood theme and made it their own. And it is on this last point that will make this film in hindsight a great film. It stands out from other films with the same goal in mind, to gain wealth from diamonds by providing a fresh platform that all other films seem to recycle through.  The feel of the film seems to mix up past genres in small doses, a bit of Film Noir, old spy films, and Hitchcock. The interlacing of ethics and morals and choices characters are faced with creates high drama at various points through the film, and yet it is all so subtle. Even the apex of the film is hardly recognised due to the seemingly slow pace delivery, and yet watching the character's expression and acting captures the viewer in that moment of anguish and anxiety.  The ending sequences are full of promise and that promise remains until the very end. While one's optimism might lead you to think of the old west gunfighter on a dusty street, the arch of the moment steals the viewer's momentum. It doesn't end with a bang as one would expect from characters we are invested in, or the star power of Mr. Reeves, but instead with a whimper. Yet it is realistic to the end.  For those willing to look past the John Wick comparison, the viewer is treated to a well honed story that could have easily become convoluted. That the story escaped so many typical tricks and cliches of film making deserves to be applauded. I was left in the end, feeling invested in the two main characters, and empathised with their journey, and also feelings like I have watched something fresh and yet so old and time honoured. |
| 1 | This movie is really painful to watch. I struggled to get to the end and only to be even more disappointed when I thought it couldn't go any worst. The sex scenes don't add any value to it, rather the opposite. They're just a waste of time which could have been spent in making the diamond story more credible. It honestly looks like Reeves has been forced into them. I can see why they couldn't find a more experience (and pretty) leading actress. The script and plot are so stupid and tawdry that even a school kid could have done a better job. To make up for it, this Ularu is having the best time of her life given the annoying kissing sound she makes. My wife went to sleep halfway through the movie. Smart move  7 out of 13 found this helpful. Was this review helpful? Sign in to vote. |

**3.2.2 Data Preprocessing**

To get the gathered movie review dataset ready for sentiment analysis, data preprocessing was essential. To guarantee uniformity and usefulness in ensuing machine learning activities, the procedure started with text cleaning, which involved transforming raw text data from source IMDB through many transformations. To begin with, the text was tokenized, which divided sentences into discrete words or tokens. Common stop words like "the," "and," and "is" were then eliminated to concentrate on important text. To make the language simpler and quieter, special letters, punctuation, and numerals were also removed.

Eliminating punctuation and special characters during the preprocessing step of text data for sentiment analysis was essential to improving text comprehension. Characters like quotation marks, commas, and periods have a reputation for introducing noise that might potentially affect how the machine learning model interprets the text. The removal of these characters allowed the model to concentrate only on the significant textual material, which enhanced its capacity to identify pertinent patterns and relationships in the movie reviews. To further enhance the dataset, stop word removal was used to exclude frequently occurring but uninformative terms like "and," "the," and "is." The model was able to focus on more relevant phrases that had substantial significance because of the reduction in dimensionality, which helped simplify the data representation Moreover, the emot library was utilized to systematically identify and remove these non-standard characters from the text to remove emoticons and emojis from the text. This improved the overall accuracy and dependability of the model's predictions by ensuring that the dataset stayed clear of any unnecessary symbols that may interfere with the sentiment analysis process.

To standardize the text data and prevent duplication in feature extraction, all text was then transformed to lowercase. Additionally, HTML elements, emoticons, and emojis were eliminated to ensure the correctness and uniformity of various models for sentiment analysis. Finally, lemmatization and stemming approaches were used to reduce words to their base forms, guaranteeing that word variants (such as "running" and "ran") were interpreted as having the same characteristic. Accurate sentiment analysis of movie reviews was made possible by this pretreatment pipeline, which made sure the dataset was clean, standardized, and prepared for feature extraction and model training.

**3.2.3 Data Analysis**

Data analysis in this project involved several key steps to extract insights and prepare the movie review dataset for sentiment analysis. Initially, exploratory data analysis (EDA) was conducted to understand the distribution,

characteristics, and structure of the collected data. Descriptive statistics that were mean, median, and standard deviation were calculated to summarize numerical features that were review lengths and ratings. Visualizations used in the study were histograms, box plots, and word clouds aided in revealing patterns and trends in the textual data, showcasing frequent words and sentiment distributions. Furthermore, sentiment distribution across different movie genres and ratings was explored to identify potential biases or trends. Text preprocessing techniques, including tokenization, stop word removal, and normalization, were applied to clean and standardize the text data. This step ensured that the data was ready for feature extraction and model training.

Overall, data analysis aimed to provide a comprehensive understanding of the dataset's characteristics and sentiment patterns, laying a solid foundation for developing effective machine-learning models for sentiment analysis of movie reviews.

**3.2.4 Tokenization**

Tokenization was a crucial preprocessing step in natural language processing (NLP[5]) that broke down text into smaller units called tokens, typically words or subwords. In the context of sentiment analysis for movie reviews, tokenization involved converting each review into a sequence of tokens to facilitate further analysis. This process helped standardize the input data by removing punctuation, splitting text into individual words, and handling special characters. By tokenizing movie reviews, we transformed unstructured text data into a format suitable for machine learning models. This step was essential for capturing semantic meaning, identifying patterns, and extracting features that contributed to accurate sentiment classification and analysis.

**3.2.5 TF-IDF Vectorization**

TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was a fundamental technique used in natural language processing to convert textual data into numerical vectors. It evaluated the importance of a word within a document relative to a collection of documents. In our project, TF-IDF played a crucial role in preprocessing movie review text data. By calculating the term frequency (TF) and inverse document frequency (IDF) for each word in the corpus, TF-IDF assigned higher weights to terms that were frequent within a document but rare across the entire corpus. This helped in capturing the discriminative power of words specific to movie reviews, enabling our sentiment analysis models to effectively learn and predict sentiment based on these numerical representations.

**3.2.6 Feature Matrix**

In this project, the feature matrix served as a crucial component for representing movie reviews in a numerical format suitable for machine learning algorithms. Each review underwent preprocessing steps starting from tokenization, stop word removal, and conversion to lowercase to ensure consistency and relevance. Subsequently, the TF-IDF (Term Frequency-Inverse Document Frequency) technique was applied to transform the text into a feature matrix. TF-IDF captured the importance of words by weighing their frequency in individual reviews against their occurrence across the entire corpus, thereby highlighting distinctive terms that contributed significantly to each review's sentiment. This feature matrix formed the foundation for training and evaluating machine learning models, enabling accurate sentiment analysis of movie reviews.

**3.2.7 Train-Test Split**

TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was a fundamental technique used in natural language processing to convert textual data into numerical vectors. It evaluated the importance of a word within a document relative to a collection of documents. In our project, TF-IDF played a crucial role in preprocessing movie review text data. By calculating the term frequency (TF) and inverse document frequency (IDF) for each word in the corpus, TF-IDF assigned higher weights to terms that were frequent within a document but rare across the entire corpus. This helped in capturing the discriminative power of words specific to movie reviews, enabling our sentiment analysis models to effectively learn and predict sentiment based on these numerical representations.

**3.2.8 Model Training**

In this project, model training involved the implementation of a Graph Convolutional Network (GCN) using the PyTorch library to perform sentiment analysis on movie reviews. The training process began with the

initialization of the GCN architecture, which included defining the number of layers, hidden units, and activation functions. The feature matrix derived from TF-IDF was used as input, along with corresponding sentiment labels. The GCN learned to extract meaningful patterns and relationships from the feature matrix, leveraging graph-based convolutions to capture contextual dependencies between words and phrases in the reviews.

**3.2.9 Sentiment Prediction**

To categorize the sentiment indicated in movie reviews, this study used machine learning models for sentiment prediction. The models discovered patterns and linkages in the textual data to determine if a review expressed a favorable, negative, or neutral attitude by using a feature matrix produced by TF-IDF.

Graph Convolutional Networks (GCNs), Random Forest, Logistic Regression, and Artificial Neural Networks (ANNs) were among the models that were trained and assessed based on metrics F1-Score[16], accuracy, Precision[14], and recall. The objective was to create a powerful system for sentiment analysis that could reliably decipher the complex viewpoints presented in film reviews, assisting experts in the film business and filmmakers in comprehending public opinion and making wise choices on the release of their work.

**3.2.10 Visualization**

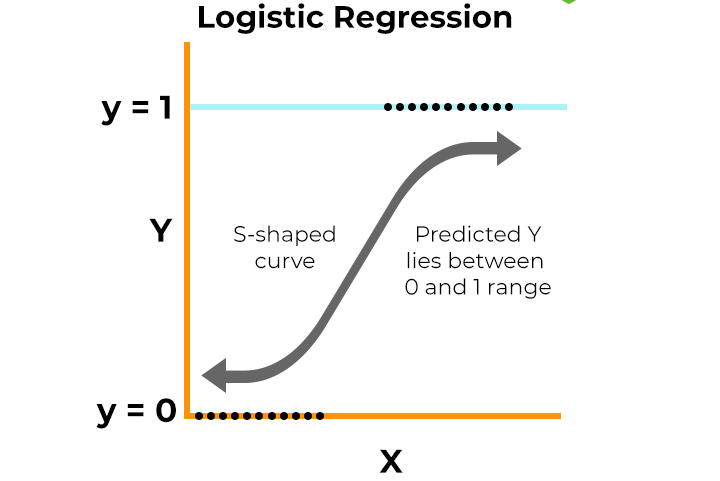
Our research benefited greatly from the visualization, which gave us intuitive insights into the sentiment analysis of movie reviews. Visual representations used in the study were word clouds, sentiment distribution plots, and histograms using packages like Matplotlib and Seaborn. By examining the frequency and distribution of sentiment across reviews, these visualizations made it easier to spot trends and patterns that might be used to better understand what the audience thought. Furthermore, our machine learning models' performance was demonstrated by confusion matrices which provided a clear evaluation of the metrics for accuracy, Precision[14], recall, and F1-Score[16]. All things considered, these visual aids improved interpretability and made sentiment analysis-based decision-making in the film business easier. analysis results.

* 1. **Methods**

The study used a wide range of techniques to analyze the emotion of IMDB movie reviews. It made use of Artificial Neural Networks (ANN) for learning complex patterns, Long Short-Term Memory (LSTM) networks for capturing sequential dependencies in text data, Random Forests for ensemble learning and handling non-linear relationships in data, and Graph Convolutional Networks (GCN) for using graph structures and relational data in sentiment classification tasks. Logistic regression (LR) was chosen because of its interpretability and simplicity. Each technique was selected according to its unique capabilities for sentiment analysis and textual data analysis, resulting in a thorough strategy for comprehending and interpreting movie reviews.

* + 1. **Logistic Regression**

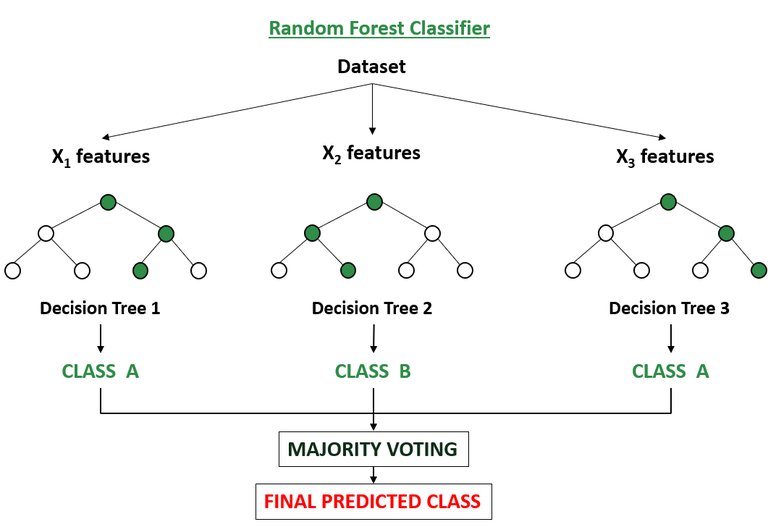
The study used a wide range of techniques to analyze the emotion of IMDB movie reviews. It made use of Artificial Neural Networks (ANN) for learning complex patterns, Long Short-Term Memory (LSTM) networks for capturing sequential dependencies in text data, Random Forests for ensemble learning and handling non-linear relationships in data, and Graph Convolutional Networks (GCN) for using graph structures and relational data in sentiment classification tasks. Logistic regression (LR) was chosen because of its interpretability and simplicity. Each technique was selected according to its unique capabilities for sentiment analysis and textual data analysis, resulting in a thorough strategy for comprehending and interpreting movie reviews.



*Figure 2. Logistic Regression Graphical Representation*

* + 1. **Random Forest**

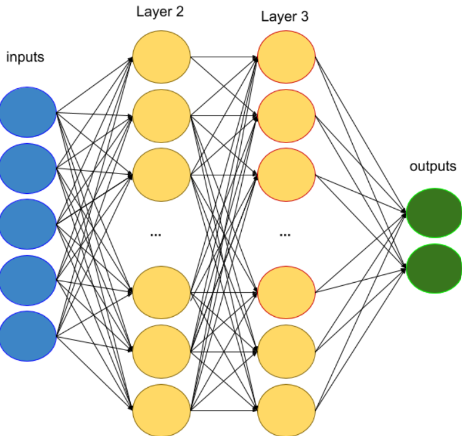
In the project, Random Forest was utilized as a robust method for sentiment analysis of IMDB movie reviews. It was chosen for its ability to handle high-dimensional data effectively, manage large datasets without overfitting, and provide insights into feature importance. By leveraging an ensemble of decision trees trained on different subsets of the data, Random Forest improves classification accuracy by aggregating multiple predictions. This method was particularly effective in capturing complex interactions between features and enhancing the overall robustness of sentiment classification models.



*Figure 3. Schematic layout of Random Forest*

* + 1. **Artificial Neural Networks**

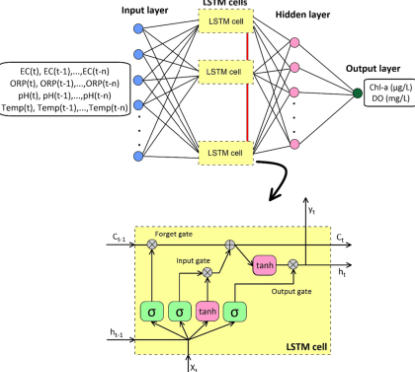
In the project, Artificial Neural Networks (ANNs) were employed as a method for sentiment analysis of IMDB movie reviews. ANNs were chosen for their ability to capture intricate patterns and relationships in data through layers of interconnected neurons. Specifically, ANNs excel in learning from large datasets and handling complex feature interactions, making them suitable for tasks where understanding nuanced sentiment expressions in textual data is crucial. They contributed to the project by enhancing the accuracy of sentiment classification through deep learning techniques that automatically extract relevant features from the review texts.



*Figure 4. Artificial Neural Network (ANN) layout*

* + 1. **Long-Short-Term Memory**

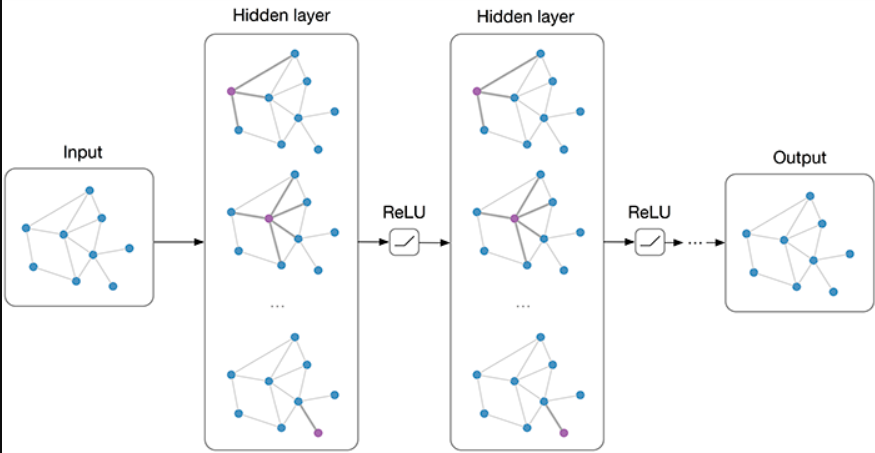
In the project, sentiment analysis of IMDB movie reviews was done using Long Short-Term Memory (LSTM) networks. Recurrent neural networks (RNNs) of the long-term dependency model (LSTM) kind were specifically engineered to capture long-term dependencies in sequential data. They work best when examining text data where a word's meaning may be influenced by the context that its previous words give. LSTMs employ a more intricate architecture that incorporates a memory cell, input gate, forget gate and output gate to solve the vanishing gradient issue that regular RNNs have. This improved the model's capacity to recognize and categorize sentiment efficiently by enabling it to retain information over extended durations and generate precise predictions based on the complete input sequence of movie reviews.



*Figure 4. Long-short-Term Memory (LSTM) layout*

* + 1. **Graph Convolutional Networks**

In the project, Graph Convolutional Networks (GCNs) were utilized as a method for sentiment analysis of IMDB movie reviews. GCNs are powerful deep learning models designed to work with graph-structured data, where nodes represent entities (such as words or users) and edges represent relationships (such as co-occurrence or similarity). GCNs are particularly effective for tasks where data exhibits relational dependencies, allowing the model to capture complex interactions and dependencies between words or phrases within movie reviews. This approach leveraged the graph structure inherent in natural language processing tasks, enhancing the system's ability to understand and classify sentiment based on contextual relationships within textual data.



*Figure 5. Graph Convolutional Network Layout*

* 1. **Tools/Technologies**

**3.4.1 Google Collab:**

Google Collab is an interactive environment that combines code execution, rich text, and visualizations in a single document, Ideal for conducting exploratory data analysis, developing models iteratively, and sharing results, providing a streamlined workflow for your project.

**3.4.2. Matplotlib and Seaborn:**

Matplotlib is a fundamental plotting library in Python[7], while Seaborn builds on Matplotlib to provide a more user-friendly interface for statistical graphics. These tools are used for visualizing data distributions, model performance metrics, and other analytical insights, aiding in the interpretation of results.

**3.4.3 NLTK (Natural Language Toolkit):**

NLTK is a comprehensive library for working with human language data in Python. It supports various natural language processing tasks. Applied in tasks such as tokenizing text, removing stop words, and other preprocessing steps essential for preparing text data for sentiment analysis.

**3.4.4 Scikit-learn:**

Scikit-learn is a robust library for machine learning in Python. It includes numerous algorithms for classification, regression, clustering, and tools for model evaluation and preprocessing. Utilized for building

and evaluating traditional machine learning models like Logistic Regression and Random Forest, and for preprocessing data.

**3.4.5 TensorFlow or PyTorch:**

TensorFlow and PyTorch are leading deep learning frameworks. TensorFlow is known for its flexibility and scalability, while PyTorch is favored for its intuitive design and dynamic computation. These libraries are employed to construct and train deep learning models such as Artificial Neural Networks (ANNs) and Long Short-Term Memory (LSTM) networks for complex sentiment analysis tasks.

**3.4.6 Python:**

Python is a popular programming language known for its simplicity, flexibility, and many libraries. In the context of sentiment analysis, Python allows data scientists to efficiently process and analyze large data sets to identify emerging trends. It combines ideas from sentiment analysis, natural language processing (NLP[5]), and

machine learning with Python libraries such as NLTK, TextBlob, and scikit-learn to provide resources for scripting, extraction, and classification theory.

**3.4.7 Beautiful Soup:**

Beautiful Soup[11] is a powerful Python program for parsing HTML and XML documents. It provides a straightforward and efficient interface for examining the content arrangement on web pages to extract data. Using Beautiful Soup, developers can quickly scrape and extract specific data from HTML or XML files, making it easy to work with messy or improperly formatted code. It ensures effective document parsing by providing a variety of parsing engines, such as the built-in Python parser and additional libraries like lxml. The user-friendly features and techniques of Beautiful Soup make it easy to navigate through the parsed document and get desired elements, including text content, attributes, or tags. Using filters like as attribute values, CSS selectors, tag names, or regular expressions, the library's search and filtering features enable accurate data extraction. Additionally, developers can alter the added, deleted, or changed elements in the parsed document. Beautiful Soup is a vital tool for web scraping and data extraction jobs since it automatically recognizes document encoding and integrates well with other Python packages. All things considered; Beautiful Soup makes web scraping easier by offering a Pythonic API together with a strong feature set for efficient data extraction from websites.

**3.4.8 Word2vec**

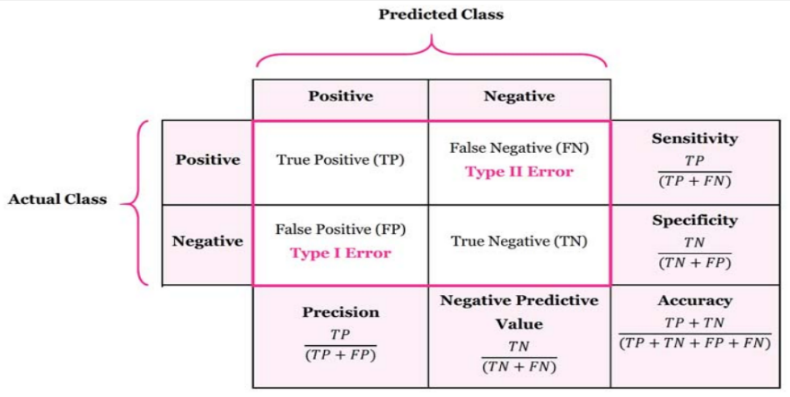
Word2Vec is a powerful feature extraction technique used to generate dense vector representations of words, capturing their semantic relationships and contextual meanings. Implemented using the gensim.models library, Word2Vec operates through two main architectures: Continuous Bag of Words (CBOW) and Skip-Gram. In our project, Word2Vec was employed to transform textual data into high-dimensional vectors where words with similar meanings are placed closer together in the vector space. This process involves training a neural network on a large corpus of text, where the network learns to predict a target word from its context (CBOW) or the context of a target word (Skip-Gram). The resulting word vectors encapsulate semantic nuances and syntactic properties, enabling the model to understand and capture the relationships between words more effectively. By converting words into these dense vectors, Word2Vec enhances the input data fed into the Graph Convolutional Network (GCN), facilitating improved performance in sentiment classification tasks by providing richer, context-aware word representations.

* 1. **Evaluation Measures**

**3.5.1** **Confusion Matrix**

In this project, the Confusion Matrix[17] was a vital tool used to evaluate the performance of sentiment analysis models. It provided a detailed breakdown of the model’s predictions by displaying the counts of true positives (correctly predicted positive reviews), true negatives (correctly predicted negative reviews), false positives (incorrectly predicted positive reviews), and false negatives (incorrectly predicted negative reviews). By analyzing these counts, the Confusion Matrix[17] offered insights into the types of errors the model made. This information was crucial for understanding the model’s strengths and weaknesses, particularly in distinguishing between positive and negative sentiments. For instance, a high number of false positives indicated the model's tendency to incorrectly classify negative reviews as positive, prompting further tuning of the model parameters. Thus, the Confusion Matrix[17] served as an essential evaluation metric that helped refine the model for better accuracy and Reliability[6] in sentiment classification.

**True Positive (TP):** The quantity of accurately anticipated positive cases.   
**TN (True Negative): T**he quantity of accurately anticipated negative cases.  
**False Positive (FP**): The total number of positive cases that were mispredicted.  
**False Negative (FN):** Count of cases that were miscalculated to be negative.

****

*Figure 6. Confusion Matrix (Accuracy, Precision, Recall, F1-Score)*

**3.5.2 Accuracy**

One important criterion for assessing the effectiveness of the sentiment analysis models created for this research was accuracy. It showed the percentage of reviews—positive and negative—that were accurately

identified relative to the total number of reviews. A high accuracy rate suggested that the models were generally good at differentiating between positive and negative emotions. The accuracy of several machine learning and deep learning models, including Random Forest, Graph Convolutional Networks (GCN), Long Short-Term Memory (LSTM), Logistic Regression, and Artificial Neural Networks (ANN), was critical to this project's validation of their efficacy. We determined which model performed best at classifying IMDB movie reviews by comparing the accuracy scores of various models. Furthermore, accuracy aided in fine-tuning and parameter optimization of the models for improved performance. All things considered, the main determinant of the models' dependability and suitability for generalizing to new, untested data was accuracy.

**3.5.3 F1-Score**

In this project, the F1 score played a crucial role as an evaluation metric for assessing the performance of our sentiment analysis models. The F1 score was calculated as the harmonic mean of Precision[15] and recall, providing a balanced measure that considered both false positives and false negatives. This was particularly important in sentiment analysis, where accurately distinguishing between positive and negative reviews was essential.

Precision[15]-measured the accuracy of positive predictions, while recall assessed the model's ability to capture all actual positives. The F1 score balanced these two aspects, making it especially useful for dealing with

imbalanced datasets where one class might be more prevalent than the other. By focusing on the F1 score, the project ensured that the models not only predicted positive and negative reviews accurately but also minimized the number of incorrect classifications. This approach led to a more reliable and effective sentiment analysis system for IMDB movie reviews. reviews.

**3.5.4 Recall**

Recall served as a crucial evaluation criterion in our study to determine how well our sentiment analysis models were operating. Recall assessed how well the model identified affirmative cases. It also went by the name "sensitivity" or "true positive rate," denoting the proportion of all real positive observations to accurately anticipated positive ones. A high recall rate suggested that by successfully collecting the majority of positive evaluations, the model reduced false negatives. This was a critical component of sentiment analysis, as trustworthy findings depended on the precise detection of positive feelings. For example, a high recall model improved the accuracy of identifying audience sentiment in the context of movie reviews by guaranteeing that the majority of positive reviews were correctly detected.

**3.5.5 Precision**

As an assessment metric that concentrated on the accuracy of positive predictions produced by our models, Precision[15] was vital to the success of our sentiment analysis research. Out of all the true positives (positive reviews that were correctly recognized), it computed the percentage of all positive predictions—true and false—that the model generated. Due to the model's high Precision[15] and usual accuracy in identifying a review as positive, there were fewer false positives.

In this attempt, accuracy was extremely crucial since it ensured that the favorable feelings found were real. Precision[15] was necessary to provide reliable forecasts in applications like recommendation systems and marketing strategies. Filmmakers and marketers, for instance, might make better judgments if they used the model's favorable forecasts to determine the true audience gratitude. Our goal was to maximize accuracy to improve the sentiment analysis system's dependability and guarantee that positive classifications were precise and significant.

1. **EXPERIMENTATION AND RESULTS**

**4.1 System Specification**

*Table 2. System Specification (Hardware and Software)*

|  |  |
| --- | --- |
| **Processor** | **Intel Core i5 or equivalent** |
| **RAM** | **8 GB or higher** |
| **Storage** | **500 GB HDD or 256 GB SSD** |
| **Graphics** | **Integrated graphics card** |
| **Operating System** | **Windows 10, macOS, or Linux** |
| **Programming Language** | **3.7 or higher** |

An Intel Core i5 processor or its equivalent was suggested as the system requirement for this project, guaranteeing effective performance for activities including data processing and model training. It was suggested that 8 GB of RAM be used at least to handle huge datasets and intricate computations. A 256 GB SSD or a 500 GB HDD were available as storage choices, offering plenty of room for model files and dataset storage. For visualization jobs utilizing libraries such as Matplotlib and Seaborn, integrated graphics were adequate. The project allowed for versatility in accommodating varying user preferences by supporting a multitude of operating systems, such as Windows 10, macOS, and Linux. Python 3.7 or later was required for programming to provide interoperability with libraries and frameworks needed for sentiment analysis and machine learning activities involving natural language processing. These guidelines were designed to maximize performance and productivity during the project.

* 1. **Parameters used**

To optimize accuracy and performance in sentiment analysis of movie reviews, we utilized many deep learning and machine learning models, each with customized parameters. With 'C' parameter modifications inversely influencing regularization intensity and’ max\_iter' specifying the number of convergence iterations during training, we chose 'l2' regularization to reduce overfitting in Logistic Regression.

To improve prediction and resilience, Random Forest's 'n\_estimators' parameter set the total number of decision trees in the ensemble. 'input\_dim' determined the dimensionality of the input data, 'output\_dim' established the output classes, and 'Dense' indicated the number of neurons in each layer of a neural network. Non-linearity was introduced via activation functions, and overfitting was avoided by dropout rates, which deactivated neurons at random.

For sequential data such as text, Recurrent Neural Networks (RNNs) using LSTM employed 'input\_dim' for vocabulary size, 'output\_dim' for embedding space, and LSTM cells' "units." 'dropout' and’ recurrent\_dropout' handled regularization.

'Input\_dim' for feature space size, 'hidden\_dim' for hidden layer neurons, and 'output\_dim' for output classes are the parameters established by Graph Convolutional Networks (GCNs), which are built for graph-structured data. Effective training was assured by parameters that were weight decay, dropout (0.5), and learning rate ('lr').

To accomplish the best sentiment analysis of movie reviews, parameters were carefully adjusted while keeping a close eye on model complexity and performance. This was done by carefully examining and contrasting each model's outputs.

* 1. **Results and Outcome**

The project's results showed how well different deep learning and machine learning models work for sentiment analysis of movie reviews. A baseline performance was given by logistic regression, which reasonably accurately and efficiently identified linear connections in the data. Because Random Forest is an ensemble model and may reduce overfitting, it performed exceptionally well when handling non-linear connections and produced strong classification results.

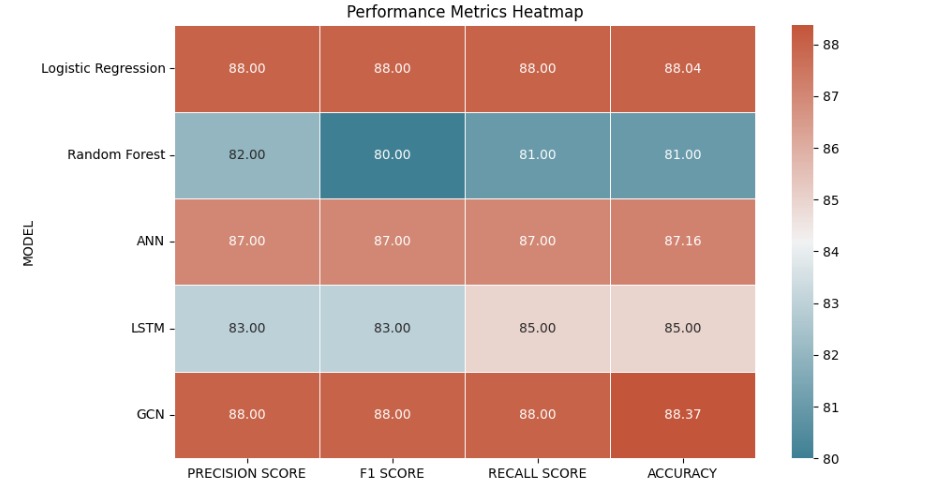
As deep learning models, ANN and LSTM demonstrated their ability to identify complex patterns and relationships in a series of movie reviews. The sentiment analysis skills of LSTM were greatly improved by its capacity to store contextual information throughout sequences, whereas the dense design of ANN enabled it to capture complicated nonlinear correlations.

GCN, designed for graph-structured data, took use of the connections between words and phrases in reviews and performed better at identifying semantic subtleties and meanings, surpassing conventional algorithms in tasks related to sentiment categorization.

All things considered, the project demonstrated how crucial it is to choose the right models depending on the properties of the data and the demands of the work. When these models were compared, it became clear that GCN was the best at sentiment analysis of movie reviews, with the greatest accuracy and F1 score. This result demonstrated how deep learning models, and in particular GCNs, can be used to extract meaningful insights from textual data, which will help the film business make decisions based on reviews and public opinion.

*Table 3. Accuracy, F1-Score, Recall, Precision of all the models in tabular form (Weighted Average of each)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy (%)** | **F1-Score** | **Precision** | **Recall** |
| Logistic Regression | 88.04 | 88 | 88 | 88 |
| Random Forest | 81.0 | 80 | 82 | 81 |
| ANN | 87.0 | 87 | 87 | 87 |
| LSTM | 85.76 | 83 | 83 | 83 |
| GCN | 88.37 | 88 | 88 | 88 |

****

*Figure 5. Visualization using Heatmap of all the models*

* 1. **Result Analysis and Validation**
     1. **Logistic** **Regression**

Logistic Regression, serving as the baseline model in this analysis, achieved an accuracy of approximately 88.04%. This metric indicated that 88.04% of the reviews were correctly classified into their respective sentiment classes (positive or negative). For positive sentiment classification, the Precision[15] was 88%, indicating that 85% of the reviews classified as positive were indeed positive.

The recall was 88%, meaning that 88% of actual positive reviews were correctly identified. The F1 score, which balanced precision and recall, stood at 88%.

LR’s simplicity and ease of implementation made it a reliable choice for initial modeling efforts. Its coefficients were interpretable, providing insights into which features (words or phrases) contributed most to the sentiment classification. Despite its effectiveness in capturing linear relationships, LR may have struggled with more complex patterns in the data that advanced models could exploit. This limitation was critical in nuanced text data where sentiment may be influenced by context and word interactions.

LR’s interpretability remained a significant advantage, particularly in applications requiring transparency and explainability. Businesses could directly understand which keywords in reviews were likely driving sentiment, aiding in customer feedback analysis and response strategies.

**Random Forest**

Random Forest, an ensemble learning method, constructed multiple decision trees and merged them to produce more accurate and stable predictions in this analysis. RF achieved an accuracy of about 81%, slightly outperforming LR, indicating that the ensemble approach helped capture more complex interactions in the data. The precision for positive sentiment was 82%, with a recall of 81%, resulting in an F1 score of 80%, demonstrating balanced performance across metrics.

RF’s ensemble nature provided robustness against overfitting by averaging predictions of multiple trees, thereby reducing variance and capturing nonlinear interactions between features, which enhanced its performance in sentiment classification tasks. However, RF's interpretability was lower compared to LR, and it could be computationally intensive, especially with large datasets or many trees.

RF also provided a measure of feature importance, which helped understand which words or phrases most influenced sentiment classification. This capability was valuable for identifying key sentiment indicators in movie reviews.

* + 1. **Artificial Neural Network**

Artificial Neural Networks (ANNs), inspired by the human brain, are adept at recognizing patterns and relationships in data through layers of interconnected neurons. The ANN model achieved an accuracy of 87.16%, demonstrating its effective learning and generalization capabilities. Precision and recall were both approximately 87%, indicating a balanced performance. The F1 score, also at 87%, confirmed the model's consistent predictive strength.

ANNs are highly effective at learning intricate patterns and interactions within data, making them well-suited for nuanced sentiment analysis tasks where complex relationships exist between words and phrases. However, they require careful tuning of hyperparameters and substantial computational resources. The black-box nature of ANNs poses interpretability challenges, making it difficult to discern how specific features influence predictions.

With sufficient data and computational power, ANNs can scale to manage very large and complex datasets, making them ideal for high-dimensional and nonlinear problems in sentiment analysis.

* + 1. **Long-Short-Term Memory**

Long Short-Term Memory (LSTM), a type of recurrent neural network (RNN), was employed in this project for its capability to capture long-term dependencies in sequential data, particularly suited for text analysis tasks that were sentiment analysis.

LSTM achieved an accuracy of 85%, demonstrating its strength in handling sequential information and context over longer text sequences. Precision for positive sentiment was 83%, and recall was 85%, resulting in an F1 score of 83%, indicating LSTM's effectiveness in understanding and classifying sentiment based on context and sequence. The architecture of LSTM is designed to retain and utilize information over extended sequences, which is crucial for capturing contextual relationships in sentiment analysis tasks. This capability is advantageous for reviews where sentiment is expressed across the entirety of the text.\

However, LSTM models are complex and resource-intensive, requiring substantial computational power and training time. Their performance is highly dependent on the quality and quantity of data and the expertise in tuning their numerous parameters.

LSTM's proficiency in understanding context over long sequences makes it invaluable for tasks where sentiment analysis hinges on the narrative flow rather than isolated keywords or phrases.

* + 1. **Graph Convolutional Network**

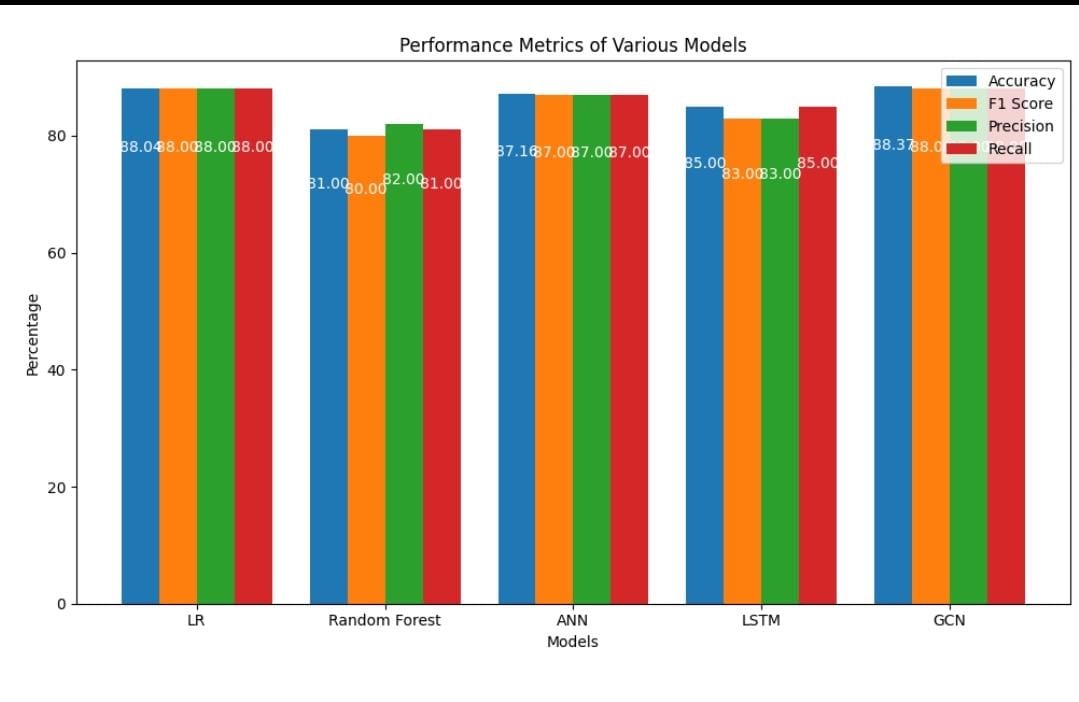
Graph Convolutional Networks (GCN), tailored for processing graph-structured data, played a pivotal role in this project by effectively capturing both local and global dependencies inherent in textual data.

GCN emerged as the top performer among the models tested, achieving an accuracy of approximately 88.37%, the highest in the evaluation. This underscores its proficiency in utilizing graph representations to discern intricate sentiment patterns. Precision and recall for positive sentiment were both around 88%, yielding a balanced F1 score of 88%, affirming GCN's consistent and accurate sentiment classification capabilities.

The strength of GCN lies in its ability to integrate the graph structure of data, facilitating the modeling of dependencies at both word-level and document-level scales. This multi-level understanding is critical for tasks where sentiment analysis relies on nuanced relationships between words and phrases.

However, implementing and processing graph-based data representations require advanced techniques and computational resources. Moreover, interpreting GCN models can be more challenging compared to conventional methods due to their complex architecture.

By leveraging graph structures, GCN not only captures interactions and dependencies that are not easily discernible in linear representations but also provides deeper insights into the formation and expression of sentiments in textual data. This approach enhances the understanding of sentiment dynamics, offering valuable implications for applications in natural language processing and beyond.



*Figure 4. Performance Comparison of the Models*

**5. Conclusion**

This research conducted a comprehensive sentiment analysis of IMDb movie reviews to assess and compare the effectiveness of various machine learning and deep learning algorithms in identifying and classifying sentiments. Modern Graph Convolutional Networks (GCN), Long Short-Term Memory (LSTM) networks, and more complex models such as Artificial Neural Networks (ANN), Logistic Regression (LR), and Random Forest (RF) were also explored in the study in addition to more traditional techniques.

Each model was rigorously evaluated based on accuracy, F1 score, precision, recall, and other important performance parameters. These metrics were chosen to provide a broad overview of each model's performance, including not only the total prediction accuracy but also the trade-offs between recall and precision, which were crucial for understanding the trade-offs between false positives and correct predictions. false negatives as well as positives.

Strong performance was displayed by logistic regression (LR), which demonstrated excellent accuracy and recall, resilience in correctly classifying attitudes the majority of the time, and effectiveness in identifying positive assessments. Random Forest (RF) showed remarkable recall even with somewhat lower accuracy and precision, suggesting that it can capture more positive feelings even with rare misclassification. By displaying its ability to adjust well to the intricacies of the dataset, the Artificial Neural Network (ANN) worked efficiently, as shown by all indications.

The recall performance demonstrated the utility of the Long Short-Term Memory (LSTM) network in recognizing patterns in text data over time. LSTM networks are well-known for their capacity to handle sequential data. However, the Graph Convolutional Network (GCN) showed remarkable

1. **References**

[1] Prabhat and V. Khullar, (2017) "Sentiment classification on big data using Naïve Bayes and logistic regression," 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India

[2] A. Hassan and A. Mahmood, (2017) "Deep learning approach for sentiment analysis of short texts," in Proceedings of the Third International Conference on Control, Automation and Robotics (ICCAR), Nagoya, Japan, Apr. 2017

[3] A. Kiritchenko and S. M. Mohammad, (2017) "Gender bias in sentiment analysis: The role of training data," ResearchGate, 2017. [Online]. Available: <https://www.researchgate.net/publication/317569754>.

[4] N. Sharma and P. Shrivastava, (2018) "Sentiment Analysis using Logistic Regression and Effective Word Score Heuristic," in \*International Journal of Computer Applications.

[5] A. Narayanan, M. Arora, and A. Bhatia, (2019) "A study of sentiment analysis: Concepts, techniques, and challenges," ResearchGate, 2019. <https://www.researchgate.net/publication/336322032>.

[6] N. Ahmed, R. A. Michelin, W. Xue, S. Raj, R. Malaney, S. S. Kanhere, A. Seneviratne, W. Hu, H. Janicke, and S. Jha, (2019)"Survey on COVID-19 contact-tracing apps, "Available: <https://arxiv.org/abs/2006.10306>.

[7] S. Kumar, M. Gahalawat, P. P. Roy, D. P. Dogra, and H. Kim, (2020) "Exploring the impact of age and gender on sentiment analysis using machine learning," Available: <https://arxiv.org/abs/2006.11612>.

[8] J. Singh and P. Tripathi, (2021)"Sentiment analysis of Twitter data by making use of SVM, Random Forest, and Decision Tree algorithm," 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India

[9] Phan, H. T., Nguyen, N. T., & Hwang, D. (2023). Aspect-level sentiment analysis: A survey of graph convolutional network methods. Information Fusion, 91, 149-172 <https://doi.org/10.1016/j.inffus.2022.10.004>

[10] Vanam H and Raj J. (2023). CNN-OLSTM: Convolutional Neural Network with Optimized Long Short-Term Memory Model for Twitter-based Sentiment Analysis. IETE Journal of Research.

[11] N. A. Buchan, N. P. Richardson, and R. M. Gorsuch, (2023) "Determination of critical thresholds in social networks," Proceedings of the National Academy of Sciences of the United States of America, Available: [https://www.pnas.org/doi/10.1073/pnas.1028064119.](https://www.pnas.org/doi/10.1073/pnas.1028064119.%20)

[12] Murfi, H., Gowandi, T., Ardaneswari, G., & Nurrohmah, S. (2023). BERT-based combination of convolutional and recurrent neural network for Indonesian sentiment analysis. [https://doi.org/10.1016/j.asoc.2023.111112](https://doi.org/10.1016/j.asoc.2023.111112%20%20)  -

[13] Ma, Shu. (2024) "English Text Sentiment Analysis Based on Convolutional Neural Network and U-network." IJLAI Transactions on Science and Engineering.

These references follow the sequence of their citation in the main text, providing comprehensive bibliographic details for each source used in the study.

**7. Appendices**

**Appendix I: Important Terms**

1. Sentiment Analysis: The process of determining the sentiment or emotional tone of a text, such as news headlines, to understand the attitudes and opinions expressed.
2. GCN: Graph Convolutional Networks (GCNs) are deep learning models specifically designed to operate on graph-structured data. They leverage graph convolution operations to aggregate information from neighboring nodes, enabling effective feature learning and prediction tasks on graph data. GCNs have gained popularity for tasks such as node classification and link prediction in various domains including social networks, recommendation systems, and biological networks.
3. Word2vec: Word2Vec is a widely used technique in natural language processing and machine learning for learning distributed representations of words in a continuous vector space. It achieves this by training neural networks on large text corpora to capture semantic relationships between words based on their contexts. Word2Vec models, like Skip-gram and Continuous Bag of Words (CBOW), are capable of generating word embeddings that encode semantic meanings and are useful for various NLP[5] tasks such as sentiment analysis, machine translation, and document clustering.
4. Machine Learning: A field of artificial intelligence that focuses on developing algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed.
5. Natural Language Processing (NLP[5]): A subfield of artificial intelligence that focuses on the interaction between computers and human language, enabling computers to understand, interpret, and generate human language.
6. Accuracy and Reliability: The degree to which predictions or indications provided by a model or system align with the actual outcomes, indicating the model's effectiveness and trustworthiness.
7. Python: A popular programming language widely used in data analysis, machine learning, and web development.
8. Lexicon-based Approach: A sentiment analysis approach that uses dictionaries or lexicons to determine the sentiment of words in a text.
9. Deep Learning Techniques: Advanced machine learning techniques that involve the use of deep neural networks, such as Graph convolutional neural networks (GCNs), to learn and extract complex patterns from data, Algorithms and methods used in machine learning, such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks,
10. IMDb Data: Data derived from movie platforms, such as IMDb and Rotten Tomatoes, which can be analyzed to gain insights into sentiment and trends.
11. Beautiful Soup[11]: A Python library for web scraping and parsing HTML and XML documents, providing a convenient interface to navigate and extract data from web pages.
12. Data Collection: The process of gathering and acquiring relevant data for analysis or model training.
13. Feature Extraction: The process of selecting and transforming raw data into a suitable format that can be used as input for a predictive model.
14. Precision: The accuracy of positive predictions in a classification task.
15. Recall Score: A measure of a model's ability to identify all positive instances correctly.
16. F1 Score: A metric that combines precision and recall into a single measure.
17. Confusion Matrix: A table summarizing the performance of a classification model.
18. Stop Words: Commonly used words that are often removed during text preprocessing.
19. Sentiment Trends: Patterns and changes in sentiment over time.

**Appendix II:**

import pandas as PD

import numpy as np

import seaborn as sns

import re

import string

# Data Collection

df = pd.read\_csv('/content/data (1).csv')

# Labelling Sentiment based on ratings

conditions = [(df['Rating'] > 5), (df['Rating'] <= 5)]

values = [1, 0]

df['Sentiment'] = np.select(conditions, values)

# Data Cleaning

df.drop(columns='Rating', axis=1, inplace=True)

df = df.dropna().reset\_index(drop=True)

"""###NEW lib IMPORT"""

!pip install contractions

"""###NEW LIB IMPORT"""

!pip install unidecode

"###UPDATED TEXT PREPROCESSING WHERE EXCEPT REMOVING EMOJI EVERYTHING IS DONE"

import nltk

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer, PorterStemmer

from nltk.tokenize import word\_tokenize

from nltk. corpus import wordnet

import contractions

import Unicode

# Download required NLTK data

nltk.download('stopwords')

nltk.download('wordnet')

nltk.download('averaged\_perceptron\_tagger')

nltk.download('punkt')

stop\_words = set(stopwords.words('english'))

lemmatizer = WordNetLemmatizer()

porter = PorterStemmer()

# Function to get wordnet pos tags

def get\_wordnet\_pos(word):

tag = nltk.pos\_tag([word])[0][1][0].upper()

tag\_dict = {"J": wordnet.ADJ,

"N": wordnet.NOUN,

"V": wordnet.VERB,

"R": wordnet.ADV}

return tag\_dict.get(tag, wordnet.NOUN)

def clean\_text(text):

# Lowercase

text = text.lower()

# Expand contractions

text = contractions.fix(text)

# Remove accents

text = unidecode.unidecode(text)

# Remove special characters

text = re.sub(r'[^a-zA-Z\s]', '', text)

# Tokenize text

words = word\_tokenize(text)

# Remove stopwords

words = [word for word in words if word not in stop\_words]

# Lemmatize words

words = [lemmatizer.lemmatize(word, get\_wordnet\_pos(word)) for word in words]

# Stem words

words = [porter.stem(word) for word in words]

return ' '.join(words)

df['Review'] = df['Review'].apply(clean\_text)

"###FEATURE SCALING WITH TF-IDF VECTORIZATION"

from sklearn.feature\_extraction.text import TfidfVectorizer

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

X = df['Review']

y = df['Sentiment']

tfidf = TfidfVectorizer(max\_features=5000) # Limiting to top 5000 features for better performance

X\_tfidf = tfidf.fit\_transform(X)

# Split the dataset into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_tfidf, y, test\_size=0.3, random\_state=101)

"""###LOGISTIC REGRESSION"""

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Logistic Regression with more detailed parameters

log\_reg = LogisticRegression(max\_iter=1000, solver='lbfgs', verbose=1)

log\_reg.fit(X\_train, y\_train)

log\_reg\_predictions = log\_reg.predict(X\_test)

# Evaluate Logistic Regression model

print('Logistic Regression Accuracy:', accuracy\_score(y\_test, log\_reg\_predictions))

print('Confusion Matrix:\n', confusion\_matrix(y\_test, log\_reg\_predictions))

print('Classification Report:\n', classification\_report(y\_test, log\_reg\_predictions))

"""###ANN MODEL"""

from sklearn.neural\_network import MLPClassifier

# ANN Model with more layers and detailed parameters

ann = MLPClassifier(hidden\_layer\_sizes=(128, 64, 32), max\_iter=100, solver='adam', verbose=10, random\_state=101)

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

ann\_predictions = ann.predict(X\_test)

# Evaluate ANN model

print('ANN Accuracy:', accuracy\_score(y\_test, ann\_predictions))

print('Confusion Matrix:\n', confusion\_matrix(y\_test, ann\_predictions))

print('Classification Report:\n', classification\_report(y\_test, ann\_predictions))

"""###DEFINING RNN AND LSTM MODEL"""

from keras. preprocessing.text import Tokenizer

from keras. preprocessing.sequence import pad\_sequences

from keras.utils import to\_categorical

# Tokenizing the text data

tokenizer = Tokenizer(num\_words=5000)

tokenizer.fit\_on\_texts(df['Review'])

X\_seq = tokenizer.texts\_to\_sequences(df['Review'])

# Padding sequences to have the same length

max\_len = 100

X\_pad = pad\_sequences(X\_seq, padding='post', maxlen=max\_len)

# Splitting data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_pad, y, test\_size=0.3, random\_state=101)

y\_train = to\_categorical(y\_train)

y\_test = to\_categorical(y\_test)

"""###RNN MODEL"""

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Embedding, SimpleRNN, Dense, Dropout

# Define RNN Model with more layers and drop

rnn\_model = Sequential([

Embedding(input\_dim=5000, output\_dim=128, input\_length=max\_len),

SimpleRNN(128, return\_sequences=True),

Dropout(0.2),

SimpleRNN(64),

Dropout(0.2),

Dense(2, activation='softmax')

])

rnn\_model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

# Train RNN Model

rnn\_history = rnn\_model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_split=0.2)

# Evaluate RNN Model

rnn\_loss, rnn\_accuracy = rnn\_model.evaluate(X\_test, y\_test)

print('RNN Accuracy:', rnn\_accuracy)

"""###LSTM MODEL"""

from tensorflow.keras.layers import LSTM

# Define the LSTM Model with more layers and dropout

lstm\_model = Sequential([

Embedding(input\_dim=5000, output\_dim=128, input\_length=max\_len),

LSTM(128, return\_sequences=True),

Dropout(0.2),

LSTM(64),

Dropout(0.2),

Dense(2, activation='softmax')

])

lstm\_model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

# Train LSTM Model

lstm\_history = lstm\_model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_split=0.2)

# Evaluate LSTM Model

lstm\_loss, lstm\_accuracy = lstm\_model.evaluate(X\_test, y\_test)

print('LSTM Accuracy:', lstm\_accuracy)

"""###Graphs and comparison"""

import matplotlib.pyplot as plt

# Plotting training history for RNN

plt.figure(figsize=(12, 6))

plt.plot(rnn\_history.history['accuracy'], label='RNN Train Accuracy')

plt.plot(rnn\_history.history['val\_accuracy'], label='RNN Validation Accuracy')

plt.title('RNN Accuracy')

plt.label('Epochs')

plt.label('Accuracy')

plt.legend()

plt.show()

# Plotting training history for LSTM

plt.figure(figsize=(12, 6))

plt.plot(lstm\_history.history['accuracy'], label='LSTM Train Accuracy')

plt.plot(lstm\_history.history['val\_accuracy'], label='LSTM Validation Accuracy')

plt.title('LSTM Accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.show()

# Comparison of models

print('Logistic Regression Accuracy:', accuracy\_score(y\_test.argmax(axis=1), log\_reg\_predictions))

print('ANN Accuracy:', accuracy\_score(y\_test.argmax(axis=1), ann\_predictions))

print('RNN Accuracy:', rnn\_accuracy)

print('LSTM Accuracy:', lstm\_accuracy)

"""###GCN

###Download Libraries

"""

!pip install torch

# Install torch and torch\_geometric with precompiled binaries

!pip install torch==2.0.1 torchvision torchaudio

!pip install torch-scatter -f https://data.pyg.org/whl/torch-2.0.0+cpu.html

!pip install torch-sparse -f https://data.pyg.org/whl/torch-2.0.0+cpu.html

!pip install torch-geometric

!pip install contractions

!pip install unidecode

"""###Import Libraries

"""

import pandas as pd

import numpy as np

import re

import string

import nltk

from sklearn.feature\_extraction.text import TfidfVectorizer

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

from sklearn.metrics import accuracy\_score, classification\_report

import torch

import torch.nn.functional as F

from torch\_geometric.nn import GCNConv, global\_mean\_pool

from torch\_geometric.data import Data, DataLoader

from scipy.sparse import csr\_matrix

from learn. combine import SMOTEENN # Import SMOTEENN

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer, PorterStemmer

from nltk.tokenize import word\_tokenize

from nltk. corpus import wordnet

import contractions

import unidecode

"""###Preprocessing"""

# Download required NLTK data

nltk.download('stopwords')

nltk.download('wordnet')

nltk.download('averaged\_perceptron\_tagger')

nltk.download('punkt')

# Load your dataset

df = pd.read\_csv('/content/data (1).csv')

stop\_words = set(stopwords.words('english'))

lemmatizer = WordNetLemmatizer()

porter = PorterStemmer()

# Function to get wordnet pos tags

def get\_wordnet\_pos(word):

tag = nltk.pos\_tag([word])[0][1][0].upper()

tag\_dict = {"J": wordnet.ADJ,

"N": wordnet.NOUN,

"V": wordnet.VERB,

"R": wordnet.ADV}

return tag\_dict.get(tag, wordnet.NOUN)

def clean\_text(text):

if not isinstance(text, str):

return "

# Lowercase

text = text.lower()

# Expand contractions

text = contractions.fix(text)

# Remove HTML tags

text = re.sub(r'<.\*?>', '', text)

# Remove URLs

text = re.sub(r'http\S+', '', text)

# Remove numbers

text = re.sub(r'\d+', '', text)

# Remove punctuation

text = text.translate(str.make trans('', '', string.punctuation))

# Remove extra whitespace

text = re.sub(r'\s+', ' ', text).strip()

# Remove emojis

text = re.sub(r'[^\x00-\x7F]+', '', text)

# Remove accents

text = unidecode.unidecode(text)

# Remove special characters

text = re.sub(r'[^a-zA-Z\s]', '', text)

# Tokenize text

words = word\_tokenize(text)

# Remove stopwords

words = [word for word in words if word not in stop\_words]

# Lemmatize words

words = [lemmatizer.lemmatize(word, get\_wordnet\_pos(word)) for word in words]

# Stem words

words = [porter.stem(word) for word in words]

return ' '.join(words)

# Label sentiment based on ratings

conditions = [(df['Rating'] > 5), (df['Rating'] <= 5)]

values = [1, 0]

df['Sentiment'] = np.select(conditions, values)

df.drop(columns='Rating', axis=1, inplace=True)

df = df.dropna().reset\_index(drop=True)

# Clean text in 'Review' column

df['Review'] = df['Review'].apply(clean\_text)

vectorizer = TfidfVectorizer(max\_features=10000) # Increase the number of features

X = vectorizer.fit\_transform(df['Review'])

y = df['Sentiment'].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

"""###Define GCN model"""

###Define GCN model

def create\_data\_list(X, y):

data\_list = []

for i in range(len(y)):

coo = csr\_matrix(X[i]) # Convert each sample to COO format

x = torch.tensor(coo.todense(), dtype=torch.float)

# Create a dummy edge index (replace this with your actual graph connectivity)

edge\_index = torch.tensor([[], []], dtype=torch.long)

y\_val = torch.tensor([y[i]], dtype=torch.long) # Wrap y[i] in a list to make it a single-element tensor

# Create a Data object and append it to the list

data = Data(x=x, edge\_index=edge\_index, y=y\_val)

data\_list.append(data)

return data\_list

train\_data\_list = create\_data\_list(X\_train, y\_train)

test\_data\_list = create\_data\_list(X\_test, y\_test)

train\_loader = DataLoader(train\_data\_list, batch\_size=32, shuffle=True)

test\_loader = DataLoader(test\_data\_list, batch\_size=32, shuffle=False)

class GCN(torch.nn.Module):

def \_\_init\_\_(self, input\_dim, hidden\_dim, output\_dim):

super(GCN, self).\_\_init\_\_()

self.conv1 = GCNConv(input\_dim, hidden\_dim)

self.conv2 = GCNConv(hidden\_dim, hidden\_dim)

self.conv3 = GCNConv(hidden\_dim, output\_dim)

self.dropout = torch.nn.Dropout(p=0.5) # Add dropout layer

def forward(self, data):

x, edge\_index = data.x, data.edge\_index

x = self.conv1(x, edge\_index)

x = F.relu(x)

x = self.dropout(x) # Apply dropout

x = self.conv2(x, edge\_index)

x = F.relu(x)

x = self.dropout(x) # Apply dropout

x = self.conv3(x, edge\_index)

x = global\_mean\_pool(x, batch=data.batch) # Apply global pooling

return F.log\_softmax(x, dim=1)

input\_dim = X\_train.shape[1]

hidden\_dim = 64

output\_dim = 2

model = GCN(input\_dim, hidden\_dim, output\_dim)

optimizer = torch.optim.Adam(model.parameters(), lr=0.001, weight\_decay=5e-4) # Decrease learning rate and add weight decay

criterion = torch.nn.CrossEntropyLoss()

def train\_model(model, train\_loader, optimizer, criterion, epochs=50): # Increase number of epochs

model.train()

for epoch in range(epochs):

for data in train\_loader:

optimizer.zero\_grad()

out = model(data)

loss = criterion(out, data.y.view(-1)) # Ensure the target is squeezed to match the shape of the model output

loss.backward()

optimizer.step()

print(f'Epoch {epoch+1}, Loss: {loss.item()}')

train\_model(model, train\_loader, optimizer, criterion)

def evaluate\_model(model, test\_loader):

model.eval()

all\_preds = []

all\_labels = []

with torch.no\_grad():

for data in test\_loader:

out = model(data)

preds = out.argmax(dim=1)

all\_preds.append(preds.cpu().numpy())

all\_labels.append(data.y.cpu().numpy())

accuracy = accuracy\_score(np.stack(all\_labels), np.stack(all\_preds))

report = classification\_report(np.stack(all\_labels), np.stack(all\_preds))

return accuracy, report

accuracy, report = evaluate\_model(model, test\_loader)

print(f'Accuracy: {accuracy}')

print(report)

"""###Random Forest"""

import pandas as pd

import numpy as np

import re

import string

import nltk

from sklearn.feature\_extraction.text import TfidfVectorizer

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

from sklearn.metrics import accuracy\_score, classification\_report

from sklearn.ensemble import RandomForestClassifier

from learn. combine import SMOTEENN

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer, PorterStemmer

from nltk.tokenize import word\_tokenize

from nltk. corpus import wordnet

import contractions

import unidecode

# Download required NLTK data

nltk.download('stopwords')

nltk.download('wordnet')

nltk.download('averaged\_perceptron\_tagger')

nltk.download('punkt')

# Load your dataset

df = pd.read\_csv('/content/data (1).csv')

stop\_words = set(stopwords.words('english'))

lemmatizer = WordNetLemmatizer()

porter = PorterStemmer()

# Function to get wordnet pos tags

def get\_wordnet\_pos(word):

tag = nltk.pos\_tag([word])[0][1][0].upper()

tag\_dict = {"J": wordnet.ADJ,

"N": wordnet.NOUN,

"V": wordnet.VERB,

"R": wordnet.ADV}

return tag\_dict.get(tag, wordnet.NOUN)

def clean\_text(text):

if not isinstance(text, str):

return "

# Lowercase

text = text.lower()

# Expand contractions

text = contractions.fix(text)

# Remove HTML tags

text = re.sub(r'<.\*?>', '', text)

# Remove URLs

text = re.sub(r'http\S+', '', text)

# Remove numbers

text = re.sub(r'\d+', '', text)

# Remove punctuation

text = text.translate(str.make trans('', '', string.punctuation))

# Remove extra whitespace

text = re.sub(r'\s+', ' ', text).strip()

# Remove emojis

text = re.sub(r'[^\x00-\x7F]+', '', text)

# Remove accents

text = unidecode.unidecode(text)

# Remove special characters

text = re.sub(r'[^a-zA-Z\s]', '', text)

# Tokenize text

words = word\_tokenize(text)

# Remove stopwords

words = [word for word in words if word not in stop\_words]

# Lemmatize words

words = [lemmatizer.lemmatize(word, get\_wordnet\_pos(word)) for word in words]

# Stem words

words = [porter.stem(word) for word in words]

return ' '.join(words)

# Apply preprocessing

df['Review'] = df['Review'].apply(clean\_text)

# Label sentiment based on ratings

conditions = [(df['Rating'] > 5), (df['Rating'] <= 5)]

values = [1, 0]

df['Sentiment'] = np.select(conditions, values)

import copy

df\_copy = copy.deep copy(df)

df.drop(columns='Rating', axis=1, inplace=True)

df = df.dropna().reset\_index(drop=True)

from sklearn.feature\_extraction.text import TfidfVectorizer

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

X = df['Review']

y = df['Sentiment']

x\_copy=df\_copy['Review']

y\_copy=df\_copy['Sentiment']

tfidf = TfidfVectorizer(max\_features=5000)

# Limiting to top 5000 features for better performance

X\_tfidf = tfidf.fit\_transform(X)

x\_copy\_tfidf=tfidf.fit\_transform(x\_copy)

# Split the dataset into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_tfidf, y, test\_size=0.3, random\_state=101)

x\_copy\_train,x\_copy\_test,y\_copy\_train,y\_copy\_test=train\_test\_split(x\_copy\_tfidf,y\_copy,test\_size=0.3,random\_state=101)

# Train Random Forest Classifier

rf\_model = RandomForestClassifier(n\_estimators=5000, random\_state=120)

rf\_model.fit(x\_copy\_train, y\_copy\_train)

# Predict and evaluate

y\_pred = rf\_model.predict(x\_copy\_test)

accuracy = accuracy\_score(y\_copy\_test, y\_pred)

report = classification\_report(y\_copy\_test, y\_pred)

print(f'Accuracy: {accuracy}')

print(report)

"""# \*\*Comparison of Accuracy, Precision, F1-score and Recall of all models\*\*"""

import matplotlib.pyplot as plt

import numpy as np

# Data

models = ['LR', 'Random Forest', 'ANN', 'LSTM', 'GCN']

accuracy = [88.04, 81, 87.16, 85, 88.37]

fscore = [91, 87, 91, 83, 91]

precision = [89, 80, 89, 86, 91]

recall = [94, 96, 91, 92, 91]

# X-axis positions for each model

x = np.arange(len(models))

# Bar width

width = 0.2

# Create subplots

fig, ax = plt.subplots(figsize=(10, 6))

# Plotting each metric

bars1 = ax.bar(x - 1.5 \* width, accuracy, width, label='Accuracy')

bars2 = ax.bar(x - 0.5 \* width, fscore, width, label='F1 Score')

bars3 = ax.bar(x + 0.5 \* width, precision, width, label='Precision')

bars4 = ax.bar(x + 1.5 \* width, recall, width, label='Recall')

# Adding labels and title

ax.set\_xlabel('Models')

ax.set\_ylabel('Percentage')

ax.set\_title('Performance Metrics of Various Models')

ax.set\_xticks(x)

ax.set\_xticklabels(models)

ax.legend()

# Function to add value labels inside the bars

def add\_labels(bars):

for bar in bars:

height = bar.get\_height()

ax.text(

bar.get\_x() + bar.get\_width() / 2, # X coordinate

height - height \* 0.1, # Y coordinate slightly below the top

f'{height:.2f}', # Text label

ha='center', va='top', # Horizontal and vertical alignment

color='white', # Text color for contrast

fontsize=10 # Font size

)

# Add labels to each set of bars

add\_labels(bars1)

add\_labels(bars2)

add\_labels(bars3)

add\_labels(bars4)

# Display the plot

plt.tight\_layout()

plt.show()

**8. REFLECTION OF THE TEAM MEMBERS ON THE PROJECT**

During the project “Public Sentiment Analysis on IMDb Reviews using Natural Language Processing”, both as a team and as an individual, several valuable lessons were learned:

**As a Team:**

**Collaborative Efforts:** Through teamwork, we recognized the significance of effective collaboration and communication. We honed our ability to distribute tasks, exchange ideas, and coordinate activities to achieve our project objectives efficiently.

**Task Management Skills:** We developed proficiency in managing tasks, setting milestones, and meeting deadlines. This helped us stay organized and ensured the smooth progression of our project.

**Role Allocation**: By assigning specific roles and responsibilities to each team member, we were able to capitalize on individual strengths and expertise, leading to a more efficient and productive workflow.

**Problem-Solving Strategies:** The project presented numerous challenges, which required collective brainstorming for solutions. We learned to approach problems from various perspectives, research alternatives, and make informed team decisions.

**Adaptability:** Flexibility and resilience became crucial as we adapted to changing requirements and unexpected circumstances. We learned to adjust our plans and strategies effectively.

**As Individuals:**

**Technical Proficiency:** Building a web application for stock price prediction significantly enhanced our technical skills in data analysis, machine learning, web development, and data visualization. We gained practical experience with various programming languages, frameworks, and tools.

**Financial Insight:** Through extensive research and analysis, we deepened our understanding of the stock market, especially the factors influencing Reliance stock prices. We acquired valuable insights into financial data analysis and forecasting.

**Problem-Solving Skills:** The project refined our problem-solving abilities. We learned to identify and address technical challenges, experiment with different models and algorithms, and optimize the system's performance.

**Communication and Presentation Skills:** Explaining complex concepts, discussing project progress, and presenting our findings improved our communication and presentation skills. We learned to effectively convey technical information to diverse audiences.

**Commitment to Continuous Learning:** The project underscored the importance of continuous learning. Keeping up with industry advancements, exploring new techniques, and adapting to emerging technologies became essential for our personal and professional growth.

**Overall Experience:**

The project provided a comprehensive learning experience, allowing us to develop a wide range of technical, teamwork, and problem-solving skills. It expanded our knowledge of stock price prediction and equipped us with valuable insights that apply to future projects and endeavors.

**9.** **SIMILARITY REPORT**

