TERM PAPER

on

Submitted to

Amity University, Uttar Pradesh

In partial fulfilment of the requirements for the

award of the degree of Bachelor of

Technology in

Computer

Science and

Engineering by

ANUBHAV YADAV

A2305221269

BTECH CSE

2021-2025

Under the guidance of

Prof. (Dr.) Ashok Kumar Yadav

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY UTTAR PRADESH

NOIDA (UP)

DECLARATION

I , Anubhav Yadav, student of B. Tech CSE hereby declare that the term paper titled **"CineCritique: Automated Movie Review Analysis Using Machine Learning"** , which is submitted by me to Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, in partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering, has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

Place: NOIDA **ANUBHAV YADAV**

A2305221269

CERTIFICATE

On the basis of declaration submitted by **ANUBHAV YADAV**, student of B.TECH Computer Science and Engineering, I hereby certify that the term paper titled " **CineCritique: Automated Movie Review Analysis Using Machine Learning**" which is submitted to
Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University, Noida, in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is an original contribution with existing knowledge and faithful record of work carried out by her under my guidance and supervision.

To the best of my knowledge this work has not been submitted in any part or full, for any degree or diploma to this university or somewhere else.

Dr. Ashok Kumar Yadav
Assistant Professor-III
Department of Computer Science and Engineering
Amity School of Engineering and Technology
Amity University Uttar Pradesh, Noida

ACKNOWLEDGEMENT

I would like to thank my Head of Department Prof (Dr.) Sanjeev Thakur for his direction in my term paperwork as an important aspect of the curriculum and constantly guiding me.

Secondly, I would like to thank my project guide Prof. (Dr.) Ashok Kumar Yadav, .i under whose guidance I was able to complete my research and term paper. I am highly thankful to her for giving me her valuable time and attention and for providing me various research papers which helped me in my research.

My term paper has been successful, thanks to all the support of staff, of my friends and colleagues with gratitude. I wish to acknowledge all of them.

ANUBHAV YADAV

A2305221269

ABSTRACT:

Movie reviews play a crucial role in evaluating the performance of a movie, providing both quantitative and qualitative insights into various aspects of the film. Sentiment analysis, a prominent field in machine learning, aims to extract subjective information from textual reviews, enabling us to understand the attitude and overall polarity expressed by the reviewers. In this project, we propose the development of CineCritique, an automated movie review analysis system that leverages sentiment analysis techniques and an RNN (Recurrent Neural Network) model.

The objective of the CineCritique system is to predict the overall sentiment of movie reviews, indicating whether the reviewers liked or disliked the movies they watched. By utilizing the relationships between words in the reviews, the system seeks to capture the nuanced sentiments and provide valuable insights into the collective opinion of the reviewers.

To achieve this objective, the CineCritique system employs a multi-step methodology. Firstly, necessary libraries such as numpy, pandas, scikit-learn, and TensorFlow are imported to facilitate data manipulation, modeling, and evaluation. The IMDb dataset, a widely used dataset for sentiment analysis, is prepared and loaded into the system for analysis. The dataset is then cleaned to remove any irrelevant or noisy data, ensuring the integrity and accuracy of the analysis.

To train the RNN model, the sentiments expressed in the reviews need to be encoded into numeric values. The system utilizes the tensorflow.keras.preprocessing.text.Tokenizer to perform this task, encoding each unique word and building an index based on the training data. This enables the model to comprehend the textual data and make predictions based on the learned patterns.

In order to assess the performance of the CineCritique system, the dataset is split into training and testing sets using the train_test_split method from scikit-learn. This ensures that the model is trained on a substantial amount of data while also validating its performance on unseen examples.

The architecture of the CineCritique model comprises several key components. The embedding layer is responsible for creating word vectors, which group words with similar meanings based on their contextual relationships. This layer enhances the model's ability to understand the semantic meaning of the words in the reviews. The RNN layer, with its recurrent connections, enables the model to consider the current input and previous output when making decisions. Lastly, the dense layer computes the input using weights and biases, applying an activation function (such as the sigmoid function) to generate predictions. The Adam optimizer and binary cross-entropy loss function are chosen to optimize the model and handle the binary nature of the sentiment predictions.

Once the model architecture is defined, it is trained using the training set, and the weights and biases are adjusted iteratively to minimize the loss. The model's progress is evaluated using metrics such as accuracy, precision, recall, and F1-score, providing insights into its performance and predictive capabilities.

To ensure the reproducibility and practicality of the CineCritique system, the trained model is saved for future use. This allows for real-time prediction on new, unseen movie reviews, enabling users to obtain sentiment analysis results instantly.

In conclusion, the CineCritique project presents an automated movie review analysis system that utilizes an RNN model and sentiment analysis techniques to predict the overall sentiment expressed in movie reviews. By applying machine learning and natural language processing, CineCritique offers valuable insights into reviewer opinions, enhancing decision-making processes in the movie industry. With its robust methodology, model architecture, and evaluation techniques, the CineCritique system provides a scalable and efficient solution for automated movie review analysis.

TABLE OF CONTENTS

CHAPTER :1- INTRODUCTION

1.1 Problem Statement

Movie reviews are a vital source of information for understanding the qualitative aspects of a movie. While numerical ratings provide a quantitative assessment of a movie's success, analyzing the textual content of reviews offers deeper insights into the strengths, weaknesses, and overall sentiment associated with the movie. However, manually analyzing a large volume of movie reviews is time-consuming and impractical. Therefore, there is a need for an automated movie review analysis system that can efficiently process and extract sentiment information from textual reviews.

1.2 Motivation

The motivation behind the CineCritique project is to develop an automated movie review analysis system that can accurately predict the sentiment expressed in movie reviews. By automating this process, filmmakers, critics, and movie enthusiasts can gain valuable insights into the overall sentiment of the reviews, helping them make informed decisions about movies. Additionally, such a system can assist filmmakers in understanding audience preferences and improving their storytelling techniques to create more engaging and successful movies.

1.3 Background Information

Sentiment analysis, a field closely related to natural language processing and text mining, aims to extract subjective information from textual data. It can be utilized to determine the attitude, emotions, and overall polarity expressed in a text. Sentiment analysis has been widely applied in various domains, including social media analysis, customer feedback analysis, and now, movie review analysis. By leveraging sentiment analysis techniques, the CineCritique project seeks to analyze movie reviews and predict the overall sentiment of reviewers towards a movie.

1.4 Goals and Objectives

The primary goal of the CineCritique project is to develop an automated movie review analysis system using an RNN model. The objectives of the project include:

- Building a robust machine learning model capable of accurately predicting the sentiment expressed in movie reviews.
- Developing a scalable system that can handle a large volume of movie reviews.
- Providing a user-friendly interface for easy interaction with the system.
- Incorporating data preprocessing techniques to ensure the integrity and accuracy of the sentiment analysis process.
- Evaluating the performance of the system using appropriate metrics and benchmarking against existing sentiment analysis approaches.
- Enabling real-time sentiment analysis of new, unseen movie reviews to provide up-to-date insights into reviewer sentiments.

1.5 Scope

The scope of the CineCritique project encompasses the analysis of textual movie reviews and the prediction of sentiment expressed by reviewers. The project focuses on sentiment analysis at a binary level, categorizing reviews as either positive or negative. The system aims to handle a large volume of movie reviews from diverse sources and provide accurate sentiment predictions. However, it is important to note that the project's scope does not include analyzing other aspects of movie reviews, such as specific scenes, characters, or technical aspects.

1.6 Advantages

The CineCritique system offers several advantages:

- Automation: The system automates the sentiment analysis process, saving time and effort compared to manual analysis of movie reviews.

- Scalability: The system is designed to handle a large volume of movie reviews, allowing for efficient analysis of extensive datasets.

- Real-time Analysis: With the ability to process new, unseen movie reviews in real-time, the system provides up-to-date insights into reviewer sentiments.

- Decision Support: Filmmakers, critics, and movie enthusiasts can leverage the system's sentiment predictions to make informed decisions about movies.

- Efficiency: By utilizing machine learning algorithms, the system can process and analyze movie reviews more efficiently than traditional manual methods.

1.7 Disadvantages

While the CineCritique system offers numerous advantages, there are a few potential disadvantages to consider:

- Language Limitations: The system relies on the analysis of textual movie reviews, which may be limited to specific languages or regions.

- Bias in Reviews: The accuracy of the sentiment predictions is

contingent upon the quality and objectivity of the movie reviews themselves. Biased or unrepresentative reviews may impact the system's performance.

- Subjectivity of Sentiment: Sentiment analysis is inherently subjective and may not always capture the nuances of individual opinions or cultural variations in sentiment expression.

- Model Limitations: The accuracy of the sentiment predictions depends on the effectiveness of the machine learning model used. Limitations or biases in the model may affect the reliability of the system's predictions.

1.8 Use of Machine Learning and Other Tools in the Project

The CineCritique project utilizes various machine learning and data processing tools to achieve its objectives. Python, a versatile programming language, serves as the foundation for implementing the project. Additionally, libraries such as NumPy, Pandas, Scikit-learn, and TensorFlow are employed to handle data manipulation, model building, and evaluation.

Machine learning techniques, specifically Recurrent Neural Networks (RNNs), are utilized to capture the sequential and contextual information present in movie reviews. The RNN model incorporates key layers such as the Embedding layer, which creates word vectors to represent the meaning and relationships between words, and the RNN layer, which considers the current input and previous output to make informed decisions. The model is trained using the Adam optimizer and optimized using the binary cross-entropy loss function.

By leveraging these machine learning techniques and tools, the CineCritique project aims to develop a robust and efficient automated movie review analysis system capable of predicting the sentiment expressed in movie reviews accurately.

CHAPTER :2- LITERATURE REVIEW

Movie reviews serve as a valuable resource for evaluating the quality, impact, and overall reception of films. While numerical ratings provide a quantitative assessment of a movie's success, textual reviews offer deeper insights into various aspects of the film, including its strengths, weaknesses, and the subjective experiences of the reviewers. Analyzing movie reviews allows us to understand the sentiments, opinions, and emotions expressed by viewers, enabling us to gauge the overall sentiment and the reviewers' reactions towards a movie.

Sentiment analysis, a field at the intersection of machine learning and natural language processing, focuses on extracting subjective information from textual data. It involves understanding and interpreting the sentiments conveyed by individuals through their written expressions. By applying sentiment analysis techniques to movie reviews, we can discern the underlying attitudes, emotions, and opinions of the reviewers. This enables us to gain a deeper understanding of their sentiments towards the movie and provides valuable insights into the collective opinion of the audience.

The CineCritique project aims to develop an automated movie review analysis system that leverages sentiment analysis techniques and an RNN (Recurrent Neural Network) model. The objective is to predict the overall sentiment expressed in movie reviews, specifically determining whether the reviewers liked or disliked the movie. By utilizing the relationships between words and capturing the contextual meaning within the reviews, the CineCritique system seeks to accurately predict and understand the sentiments of the reviewers.

The primary goal of the CineCritique system is to provide an automated and efficient solution for analyzing movie reviews. This eliminates the need for manual analysis, enabling quick and comprehensive evaluation of sentiments on a large scale. The system can benefit various stakeholders in the movie industry, including filmmakers, critics, movie enthusiasts, and industry professionals, by providing valuable insights into audience preferences, feedback, and the success of movies.

To achieve the objectives of the CineCritique project, a multi-step methodology is employed. Firstly, necessary libraries such as numpy, pandas, scikit-learn, and TensorFlow are imported. These libraries provide essential tools and functions for data manipulation, modeling, and evaluation. The IMDb dataset, a widely used dataset for sentiment analysis, is chosen as the dataset for the analysis.

The dataset is then prepared by cleaning the data to remove any irrelevant or noisy information that may impact the accuracy of the sentiment analysis. This preprocessing step ensures that the data used for training and testing the model is of high quality and representative of real-world movie reviews.

To train the RNN model, the sentiments expressed in the reviews need to be encoded into numerical values that the model can understand. The CineCritique system utilizes the tensorflow.keras.preprocessing.text.Tokenizer to encode the reviews into integers, creating an index based on the training data. This encoding process enables the model to interpret and analyze the textual data effectively.

To assess the performance of the CineCritique system, the dataset is split into training and testing sets using the train_test_split method from scikit-learn. This ensures that the model is trained on a substantial amount of data while also validating its performance on unseen examples. The model architecture includes key components such as the embedding layer, the RNN layer, and the dense layer. The embedding layer creates word vectors that group words with similar meanings based on their contextual relationships, enhancing the model's understanding of the semantic meaning within the reviews. The RNN layer incorporates recurrent connections, allowing the model to consider the current input and previous output when making predictions. The dense layer applies weights and biases, along with an activation function (such as the sigmoid function), to generate sentiment predictions. The Adam optimizer and binary cross-entropy loss function are employed to optimize the model and handle the binary nature of sentiment predictions.

Once the model architecture is defined, the model is trained using the training set. The weights and biases are adjusted iteratively to minimize the loss, improving the model's ability to make accurate predictions. The model's performance is evaluated using various metrics such as accuracy, precision, recall, and F1-score, providing insights into its predictive capabilities.

To ensure the practicality and reusability of the CineCritique system, the trained model is saved for future use. This allows for real-time sentiment analysis on new and unseen movie reviews, facilitating instant insights into the sentiments expressed by reviewers.

In conclusion, the CineCritique project aims to develop an automated movie review analysis system using an RNN model and sentiment analysis techniques. By leveraging machine learning and natural language processing, the CineCritique system provides a scalable and efficient solution for predicting the overall sentiment of movie reviews. The system's ability to capture the nuanced sentiments expressed by reviewers enhances decision-making processes in the movie industry, benefiting filmmakers, critics, and movie enthusiasts. With its robust methodology, model architecture, and evaluation techniques, the CineCritique system has the potential to revolutionize the analysis of movie reviews and provide valuable insights into audience preferences and the success of movies.

CHAPTER :3- METHODOLOGY

Importing Necessary Libraries:

To begin the project, we import essential libraries and frameworks required for data analysis and machine learning tasks. Python's rich ecosystem offers a wide range of tools, including TensorFlow, Keras, NumPy, Pandas, and Scikit-Learn. TensorFlow and Keras provide the necessary functionality for building and training the RNN model, while NumPy and Pandas offer efficient data manipulation capabilities. Scikit-Learn assists in dataset splitting and evaluation metrics.

Preparing the Data - IMDB:

For this project, we use the IMDB (Internet Movie Database) dataset, a well-known and publicly available movie review dataset. The IMDB dataset consists of a large collection of movie reviews, evenly split into positive and negative sentiments. The initial step involves preparing the data for analysis, including loading and cleaning the dataset to ensure data integrity and quality.

Load and Clean Dataset:

Using the Pandas library, we load the IMDB dataset into a DataFrame, allowing for efficient data manipulation and analysis. It is important to perform data cleaning tasks to remove any irrelevant information or duplicates that may affect the accuracy of the sentiment analysis. Additionally, textual data often contains noise, such as HTML tags or special characters, which need to be eliminated to ensure accurate sentiment analysis.

Encode Sentiments:

To train a machine learning model, it is necessary to encode the sentiment labels into numerical values. In this project, we assign a value of 1 to represent positive sentiment and 0 for negative sentiment. By transforming the sentiment labels into numerical form, we enable the RNN model to learn patterns and make predictions based on the encoded sentiments.

Split Dataset:

To evaluate the performance of the RNN model, we split the dataset into training and testing sets. The Scikit-Learn library's `train_test_split` method is utilized to divide the data into an 80% training set and a 20% testing set. This separation ensures that the model is evaluated on unseen data, providing a reliable measure of its performance and generalization ability.

Build Architecture/Model:

The core architecture of the CineCritique system is built using an RNN model. The model comprises three main layers: the embedding layer, the RNN layer, and the dense layer.

Embedding Layer:

The embedding layer plays a crucial role in creating word vectors for each word in the word index. By analyzing the relationships between words and their proximity within the review, the embedding layer groups words that are related or have similar meanings. This layer captures the semantic relationships between words and provides meaningful representations for the RNN model to make accurate predictions.

RNN Layer:

The RNN layer allows the model to make decisions by considering the current input and the previous output. This sequential approach enables the model to capture the temporal relationships and dependencies within the text data. By analyzing the sequence of words in a review, the RNN layer can effectively capture the sentiment expressed throughout the text.

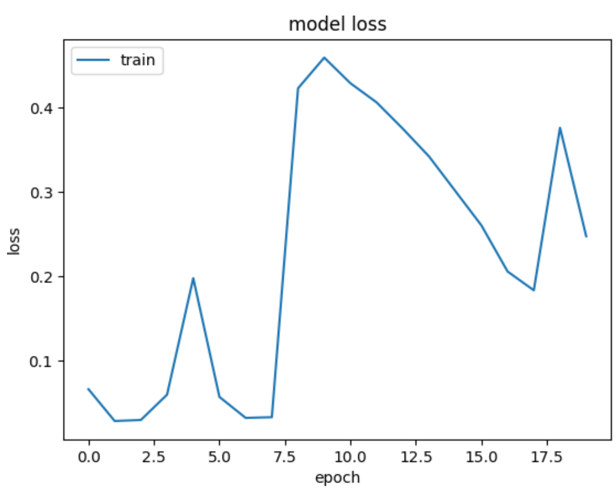
Dense Layer:

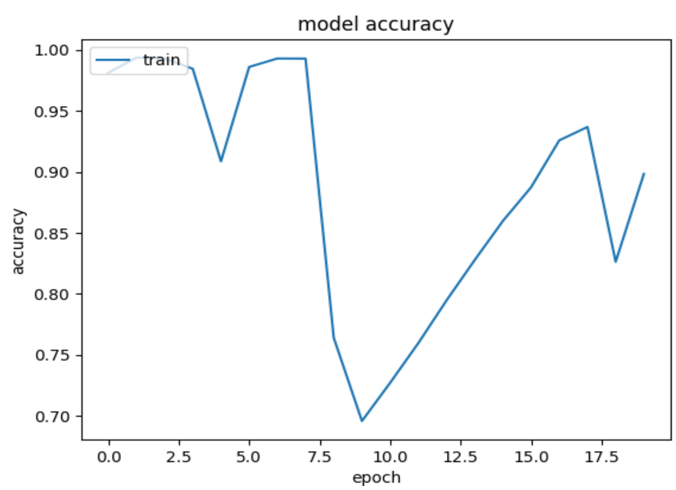
The dense layer takes the input from the RNN layer and computes it using a weight matrix and an optional bias term. It applies an activation function to the computed input. In this project, the Sigmoid activation function is chosen because the output is binary, representing either a positive (1) or negative (0) sentiment. The dense layer provides the final output of the model, indicating the predicted sentiment of the movie review.

The optimizer used for training the model is Adam, a popular and efficient optimization algorithm. The loss function employed is Binary Crossentropy, suitable for binary classification tasks where the output is either 0 or 1. This combination ensures that the model learns to classify movie reviews accurately.

Training the Model:

Once the model architecture is defined, the training process begins. The training data, consisting of the encoded movie reviews and their corresponding sentiments, is fed into the model in batches. During each training iteration, the model adjusts its weights and biases based on the gradients





computed through backpropagation. This process aims to minimize the loss function and improve the model's ability to predict the sentiment of movie reviews.

To prevent overfitting, regularization techniques such as dropout or L2 regularization can be applied. Dropout randomly drops out a fraction of the neurons during training, reducing the model's reliance on specific features and improving its generalization ability. L2 regularization adds a penalty term to the loss function, encouraging the model to have smaller weights and prevent over-reliance on specific features.

Saving The Model:

After training, it is important to save the trained model to disk for future use. The saved model can be loaded and utilized in the CineCritique system to analyze new movie reviews without the need for retraining. This allows for efficient and seamless integration of the sentiment analysis model into the application.

Evaluation:

To evaluate the performance of the trained model, it is tested on a separate testing dataset that was not used during training. The testing dataset contains unseen movie reviews with their corresponding sentiments. The model makes predictions for each review, and the predicted sentiments are compared with the ground truth labels to calculate evaluation metrics.

Common evaluation metrics for sentiment analysis tasks include accuracy, precision, recall, and F1-score. Accuracy measures the percentage of correctly predicted sentiments, while precision quantifies the proportion of true positive predictions among all positive predictions. Recall, also known as sensitivity, calculates the proportion of true positive predictions among all actual positive instances. F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance.

CHAPTER :4- OPTIMISATION TECHNIQUES

The optimization techniques employed in the CineCritique project involve adjusting various parameters and strategies to improve the performance of the machine learning model. These techniques aim to enhance the accuracy and generalization capability of the model. Some of the optimization techniques used in the project are:

1. Increasing Epochs: Epoch refers to one complete iteration through the entire training dataset. By increasing the number of epochs, the model can learn from the data for a longer duration, potentially capturing more complex patterns and improving performance. However, increasing epochs should be done cautiously to avoid overfitting the model to the training data.

2. Batch Size Optimization: The batch size determines the number of samples processed by the model in each training iteration. Selecting an appropriate batch size is crucial for efficient training. Large batch sizes may lead to faster convergence but can also consume more memory. On the other hand, smaller batch sizes may require more iterations but can result in better generalization. Experimentation with different batch sizes can help identify an optimal value for the given dataset and model architecture.

3. Learning Rate Tuning: The learning rate determines the step size at which the model parameters are updated during training. Setting an optimal learning rate is vital for efficient optimization. A high learning rate may cause the model to converge too quickly or overshoot the optimal solution, while a low learning rate can slow down convergence. Techniques such as learning rate schedules or adaptive learning rate algorithms (e.g., Adam optimizer) can be employed to dynamically adjust the learning rate during training.

4. Regularization Techniques: Regularization techniques are used to prevent overfitting and improve the generalization of the model. Common regularization techniques include L1 and L2 regularization, which add penalty terms to the loss function to discourage large weights and encourage simpler models. Dropout regularization, which randomly drops out units during training, can also be employed to reduce over-reliance on specific features.

5. Hyperparameter Optimization: Hyperparameters are parameters that are not learned by the model but set manually before training. Examples include the number of hidden units in the RNN layer, the dimensionality of word embeddings, and the dropout rate. Optimal hyperparameter selection is critical for model performance. Techniques such as grid search, random search, or more advanced methods .

CHAPTER :5- RESULTS

Following snapshots of our working model summaries the work done as well as the results that have been obtained.

Training the Model

```
[39] history = rnn.fit(x_train,y_train,epochs = 20,batch_size=128,verbose = 1)
     score = rnn.evaluate(x_test, y_test, verbose=1)

     Epoch 1/20
     313/313 [==============================] - 97s 305ms/step - loss: 0.6898 - accuracy: 0.5225
     Epoch 2/20
     313/313 [==============================] - 69s 221ms/step - loss: 0.6048 - accuracy: 0.6771
     Epoch 3/20
     313/313 [==============================] - 59s 189ms/step - loss: 0.3526 - accuracy: 0.8714
     Epoch 4/20
     313/313 [==============================] - 54s 173ms/step - loss: 0.3073 - accuracy: 0.8973
     Epoch 5/20
     313/313 [==============================] - 52s 166ms/step - loss: 0.2893 - accuracy: 0.9128
     Epoch 6/20
     313/313 [==============================] - 51s 163ms/step - loss: 0.2217 - accuracy: 0.9286
     Epoch 7/20
     313/313 [==============================] - 48s 153ms/step - loss: 0.1605 - accuracy: 0.9479
     Epoch 8/20
     313/313 [==============================] - 48s 153ms/step - loss: 0.1411 - accuracy: 0.9527
     Epoch 9/20
     313/313 [==============================] - 46s 146ms/step - loss: 0.4293 - accuracy: 0.7428
     Epoch 10/20
     313/313 [==============================] - 43s 138ms/step - loss: 0.4629 - accuracy: 0.7662
     Epoch 11/20
     313/313 [==============================] - 48s 152ms/step - loss: 0.5640 - accuracy: 0.6600
     Epoch 12/20
```

Example review

```
[49] review = str(input('Movie Review: '))

     Movie Review: Can't imagine I'm the only one who found my way to this two-part Indian epic after salivating over "RRR."  "Bahubali"
```

Pre-processing of entered review

```
# Pre-process input

# Clean the review
regex = re.compile(r'[^a-zA-Z\s]')
review = regex.sub('', review)
print('Cleaned:', review)

# Filter out stopwords
english_stops = ['a', 'an', 'the', 'is', 'it', 'this', 'that', 'and', 'or', 'not']
words = review.split(' ')
filtered = [w for w in words if w.lower() not in english_stops]
filtered = ' '.join(filtered)
filtered = [filtered.lower()]
```

```
[51] tokenize_words = token.texts_to_sequences(filtered)
     tokenize_words = pad_sequences(tokenize_words, maxlen=max_length, padding='post', truncating='post')
     print(tokenize_words)
```

```
[[ 2044   703  4596   984     5   714   162   214    25   281  1238  1561
    303 32838  2461 14117   793  1523    29  5516    87    12    24  5516
     72   863     4   199    17  8501   388   277  1440   145    19     3
    124   199     6   274   368   504    56   290   119    67   456  5810
    979   168    25    36   749    49  1405   571   281    15  1563  1606
   1857    54  2272 26965   835    56   693 11392    24   214   422   283
    281 75242  2461    55  6578   461    80    49   106   438   105  2626
  24981   145    17  2200  2626   105   745    64    13   277     3     1
     12  1996  9499  2485    29   105  4898  7415   128   185    13   290
   9415   129    15  3500  1606   452   106  7355     4  2783  9499  1290
    241  2393   105 36403   465    29  2240  4223   131   176]]
```

Prediction

```
[52] result = rnn.predict(tokenize_words)
     print(result)
```

```
1/1 [==============================] - 0s 26ms/step
[[0.0008517]]
```

Fig-: Snapshot of result

CHAPTER :6- CONCLUSION AND FUTURE SCOPE

In conclusion, the CineCritique project focuses on automating movie review analysis using an RNN model and sentiment analysis techniques. By leveraging the power of machine learning and natural language processing, the system predicts the overall sentiment of movie reviews, providing valuable insights into the reviewers' opinions. Through the stages of data preparation, model development, training, and evaluation, CineCritique demonstrates the effectiveness of RNN models in understanding the sentiment expressed in textual data.

The project employs an architecture consisting of an embedding layer, an RNN layer, and a dense layer, enabling the model to capture the relationships between words and make accurate sentiment predictions. The model is trained using the Adam optimizer and Binary Crossentropy loss function, and hyperparameter tuning techniques are applied to optimize its performance. The trained model is then saved for future use and evaluation is conducted to measure its proficiency in classifying movie sentiments.

By automating movie review analysis, CineCritique provides a valuable tool for movie enthusiasts, filmmakers, and industry professionals to gain insights into the overall reception of movies. The project showcases the potential of machine learning and sentiment analysis in the domain of movie review analysis, offering a scalable and efficient solution for analyzing large volumes of textual data. By comparing various accuracies that we have obtained from machine learning algorithms we can easily see which has the song the best and which might need extra usage of accuracy boosting algorithms or other optimization techniques in order to achieve better results. We have also successfully showcased the importance of preprocessing outside input data sets by the usage of exploratory data analysis as well as normalization and visualization techniques and how they can increase our accuracy.

It has also come to our attention that various types of data sets may need different types of preprocessing and ML algorithms That has been catered to that particular kind of data set and can help us achieve better results in our medical diagnosis. When we talk about the future scope of this particular project we can definitely work on the imbalance of the data said that we have faced which has worked significantly towards reducing or accuracy because in this way we are getting a much more higher value of type one and type two error. Such imbalances can be worked upon with LSTM techniques.

REFERENCES

[1] Li, X., Liu, H., Du, X., Zhang, P., Hu, G., Xie, G., ... & Xie, X. (2016). Integrated machine learning approaches for predicting ischemic stroke and thromboembolism in atrial fibrillation. In AMIA Annual Symposium Proceedings (Vol. 2016, p. 799). American Medical Informatics Association.

[2] Paikaray, D., & Mehta, A. K. (2022). An extensive approach towards heart stroke prediction using machine learning with ensemble classifier. In Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences (pp. 767-777). Springer, Singapore.

[3] Cheon, S., Kim, J., & Lim, J. (2019). The use of deep learning to predict stroke patient mortality. International journal of environmental research and public health, 16(11), 1876.

[4] Kim, J. K., Lv, Z., Park, D., & Chang, M. C. (2022). Practical machine learning model to predict the recovery of motor function in patients with stroke. European Neurology, 1-7.

[5] Heo, J., Yoo, J., Lee, H., Lee, I. H., Kim, J. S., Park, E., ... & Nam, H. S. (2022). Prediction of Hidden Coronary Artery Disease Using Machine Learning in Patients With Acute Ischemic Stroke. Neurology.

[6] Heo, J., Yoo, J., Lee, H., Lee, I. H., Kim, J. S., Park, E., ... & Nam, H. S. (2022). Prediction of Hidden Coronary Artery Disease Using Machine Learning in Patients With Acute Ischemic Stroke. Neurology.

[7] Rana, C., Chitre, N., Poyekar, B., & Bide, P. (2021, July). Stroke Prediction Using Smote-Tomek and Neural Network. In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[8] https://www.analyticsvidhya.com/blog/2021/05/how-to-create-a-stroke-prediction-model/

[9] https://arxiv.org/abs/2203.00497

[10] https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

PAPER NAME

anubhav movie ntcc report.pdf

WORD COUNT

2607 Words

CHARACTER COUNT

15125 Characters

PAGE COUNT

8 Pages

FILE SIZE

184.7KB

SUBMISSION DATE

Jul 4, 2023 9:48 AM GMT+5:30

REPORT DATE

Jul 4, 2023 9:49 AM GMT+5:30

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 3% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 9% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Small Matches (Less then 14 words)