National College of Ireland

AI-Enhanced Product Recommendation System using YouTube Comments Analysis

MSc Research Project
Data Analytics

Sharon George
Student ID: 21245240

School of Computing
National College of Ireland

Supervisor: Mayank Jain

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sharon George
Student ID:	21245240
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Mayank Jain
Submission Due Date:	14/12/2023
Project Title:	AI-Enhanced Product Recommendation System using YouTube Comments Analysis
Word Count:	6302
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sharon George
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☑
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☑
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☑

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI-Enhanced Product Recommendation System using YouTube Comments Analysis

Sharon George
21245240

Abstract

This project's aims to improve any AI driven product recommendations by analysing the YouTube comments and creating a user friendly Recommendation System. Traditional recommendation systems often struggle to adapt to and understand user sentiments quickly. To overcome this, we use YouTube comments as a primary data source and employ the Convolutional Long Short-Term Memory (CLSTM) algorithm for sentiment analysis. The project is unique because it uses real time YouTube comments to create a dynamic dataset for the recommendation system. The CLSTM algorithm is really good at figuring out feelings in text. It gives a lot of details about what users like and tries to make the product suggestions even better by getting more accurate and personalised. In conclusion this project uses sentiment analysis, artificial intelligence and recommendation systems to make recommendations. The CLSTM algorithm in the project performs well with an accuracy of 89%.
Keywords— AI, recommendation systems, YouTube, CLSTM, comments

1 Introduction

The main tool used in this project is the CLSTM algorithm as it is known for its ability to understand subtle emotions in any text. Choosing CLSTM emphasises the project's dedication to grasping user sentiments in YouTube comments thoroughly. These comments are filled with various opinions that hold valuable information that can help understand what a users like or dislike about different products. The main idea of the project is to use comments on YouTube as the main training data for the recommendation system. Unlike traditional systems that use fixed datasets, this project uses the real time user comment of any YouTube video. The goal is to create a recommendation system that considers the users sentiments and their preferences. The project tries to show that the CLSTM algorithm works well in understanding people's feelings in any video as explored by Mulholland et al. (2017). It aims to understand user preferences better. By using artificial intelligence to analyse YouTube comments the project aims to make a recommendation system that truly understands how users really feel and not just relying on computer algorithms. The aim is to give advice that's not just right but also connects with people personally. This project introduces an exploration of the intersection of sentiment analysis, artificial intelligence and recommendation systems. The aim is to enhance the user experience in the realm of product recommendations. In summary this project seeks to leverage the power of AI and sentiment analysis to analyse YouTube comments

and create a recommendation system that understands users' genuine feelings which will improve the overall product recommendation experience.

1.1 Motivation and Project Background

In today's online world the places where users create content like YouTube we have seen a lot more people making and watching videos. This has created a huge collection of thoughts, comments and feelings from people all over the world. If we can use these feelings in a smart way. They can be used to give us useful information for many different things like figuring out what people want or making content better. Sentiment analysis is a part of computer language understanding. It is important for figuring out and understanding these feelings from written words. It uses computer techniques to sort out and study the emotional tone, attitude or opinion in a piece of writing. This is hard because human language is tricky and depends on the situation. Recently there are many new and improved deep learning and transformer models have shown that they're really good at doing sentiment analysis. One example is the Convolutional Long Short-Term Memory (CLSTM) which is good at finding complicated language patterns and connections. Another example is the Generative Pre-trained Transformer (GPT) which is a transformer model that has changed how we understand and use computer language. These models learn a lot from big amounts of text on the internet which makes them good at understanding context and creating text that makes sense in that context. Even though these techniques are powerful using them on user-generated content from YouTube has its own challenges. Unlike carefully chosen datasets while user-made content on YouTube comes in many different writing styles, languages and levels of formality. It also often has slang, emojis and internet words that normal sentiment analysis models might not understand well. In summary this study wants to connect the best deep learning techniques with the unique problems of content made by users on YouTube. By doing this it hopes to not just improve how we do sentiment analysis but also give useful information to people who make content, market things and moderate platforms. This way like Alhujaili and Yafooz (2022) mentioned they can make better choices in the always-changing online world.

1.2 Research Question

Q1 To what extend does the CLSTM algorithm analyze sentiments in real-time YouTube comments and how does its performance compare to traditional sentiment analysis methods?
Q2 How does the proposed recommendation system incorporate sentiment analysis from YouTube comments using CLSTM in terms of accuracy and user satisfaction?

2 Related Work

2.1 YouTube Sentiment Analysis Methodologies

As YouTube is the main social media site Pradhan (2021) points out the difficulty of figuring out how people feel in the midst of so many comments. The authors proposed

using Natural Language Processing (NLP) to analyse comments on YouTube. This generates a report that categorises sentiments as positive, negative or neutral. The comments that are left on YouTube videos can have serious effects on public opinion. Therefore, understanding and processing text based data on YouTube is important. The authors introduced NLP as a tool for sentiment analysis to show its importance.

Yafooz and Alhujaili (2021) discuss ways to pick good sentiment analysis models for YouTube content. They explore different ways to do this job. They cover a range of feelings, suggesting models that go from simple (good/bad) to more categories (happy, sad, scared, surprised and angry). Acknowledges the diversity of sentiments expressed in YouTube comments and the difficulty in choosing the most accurate sentiment analysis model. Gives a sorted look that helps with studying data mining and feelings in YouTube videos made by users.

Recognising the significance of uncovering user opinions on services or products, Alhujaili and Yafooz (2021) delve into sentiment analysis methodologies applied to YouTube comments. The authors study the different ways to analyse the data and feelings. They want to find effective techniques for that can understand emotions. They say YouTube comments are important for making the content better. Alhujaili and Yafooz (2021) explore ways to understand the feelings and face the difficulty of picking the best model. Their goal is to provide insights into understanding sentiments in user-generated content.

This collection of methods displays how sentiment analysis on YouTube is changing over time. Scientists are studying problems linked to different feelings, model correctness and how comments affect public opinions. While each study is valuable there is still scope for future work to improve the methodologies and investigate the real world applications on youTube Pradhan (2021).The intersection of NLP, machine learning and user generated content presents a rich area for continued investigation.

2.2 Social Dynamics and Sensitive Topics

In Oksanen et al. (2015) studied how people interact in pro-anorexia communities on YouTube where harmful weight loss practices are promoted. They checked how the people would react to videos that disagreed with these groups and made them feel emotions. The study found that comments on videos against proanorexia had more positive feelings than comments on proanorexia videos. This shows that creating content to oppose harmful messages is powerful. This study is crucial for professionals working with young people to emphasise the importance of knowing how these content created by users can impact actually impact eating disorders.

Chakravarthi (2022) contribute to the literature by addressing abusive content on social media platforms. Unlike existing systems focusing on mitigating negativity but this study promotes free expression through positivity.The results highlight that encouraging positive talks online is a good option instead of blocking negative language. This research adds to the talk about how we handle content on the internet and brings in a new way of looking at managing online discussions. The results demonstrate that encouraging positive online interactions can be a good option instead of just blocking out all the negative comments. This research adds to the discussion about how to control content online and brings in a new way of looking at managing conversations on the internet.

These studies explore how people feel and manage content on social media. Oksanen et al. (2015) talk about how people's reactions to sensitive content can impact things. Chakravarthi (2022) share a fresh view on ways to moderate content. Oksanen et al.

(2015) talked about feelings connected to pro-anorexia content. Chakravarthi (2022) suggest a new way of moderating content that fills a gap in research. They focus on encouraging positive interactions instead of just censoring content.These studies don't exactly disagree, but they show an ongoing struggle between handling sensitive topics carefully and encouraging positive interactions. More research can look into how different ways of moderating content affect how people act. Also, figuring out how to keep positive interactions going in online spaces is something that needs more investigation.

Oksanen et al. (2015) and Chakravarthi (2022) provide valuable insights into social dynamics and sensitive topics on YouTube. These studies deepen our understanding of user reactions to pro-anorexia content and propose innovative approaches to content moderation opening avenues for further research in the evolving landscape of online interactions.

2.3 Multilingual Sentiment Analysis

Tehreem (2021) look at understanding feelings in different languages, especially in Roman Urdu, which is used for Urdu. They study the comments on YouTube about Pakistani dramas using a dataset that classifies sentiments into positive, negative or neutral. They try five different ways to learn from the data, and the Support Vector Machine (SVM) turns out to be the best, achieving a 64% accuracy. Aribowo et al. (2021) and team are dealing with the difficulties of understanding feelings in Indonesian, a language with lots of extra words, subtle meanings, and slang. They use a tree-based ensemble machine learning method to handle Cross-Domain Sentiment Analysis (CDSA). By looking at how getting rid of extra words and changing slang words affects the results, they find good CDSA models. They use different sets of YouTube comments for the study, one set without removing extra words and changing slang, and another with these changes.Fancy tree-based teamwork models, particularly Extra Tree, do better than solo classifiers, getting an impressive 91.19% accuracy. This model shows a hopeful technique understanding feelings in the Indonesian language. They point out how Roman Urdu is common in Pakistan and how the Indonesian language is pretty tricky. Two studies have explored why it's important to analyse the emotions in languages other than English such as Roman Urdu in Pakistan and Indonesian. These studies show that it is necessary to consider different languages during sentiment analysis and they highlight the challenges of handling unique words, common words and informal language in each language. The studies emphasize the importance of addressing these languagespecific challenges to get accurate results. Tehreem (2021) and Aribowo et al. (2021), along with their teams, greatly help us understand sentiment analysis in different languages. These studies go further than before by looking at Roman Urdu and Indonesian. They also talk about how different machine learning methods perform. Both studies highlight the challenges and successes in understanding emotions in languages with unique features. They give new ideas on getting rid of confusing language and give good models. This helps us understand sentiment analysis better in different languages. The studies do not conflict with each other but rather complement each other by addressing different languages and contexts. Tehreem (2021) share thoughts on understanding feelings in Roman Urdu, and Aribowo et al. (2021) offer important discoveries about understanding emotions in Indonesian. In the future researchers can study more about understanding feelings in different languages. They can look at new languages think about cultural differences and make better models

that handle language specific problems. These studies give us ways to explore feelings in languages that haven't been studied much. These insights collectively contribute to a more comprehensive understanding of the challenges and opportunities in multilingual sentiment analysis, providing a foundation for future research in this evolving field.

2.4 Innovations in Sentiment Analysis

Singh and Tiwari (2021) study uses machine learning techniques to analyse sentiments in YouTube comments related to popular topics. The goal is to study feelings to find patterns, seasonal changes and predictions. This helps in understand how the real world events affect sentiments. The research proves a clear link between what people feel and the events that are happening around them. By employing a corpus of 1500 annotated citation sentences the study preprocesses comments to remove the noise. Six machine learning algorithms, including Naïve-Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbor (KNN) and Random Forest (RF) are used for classification. Evaluation metrics such as F-score and Accuracy score are used to find the system's accuracy.

Dabas et al. (2019) introduce a system dedicated to the comprehensive analysis and classification of YouTube video comments. Comments are ingested into the Hadoop Distributed File System (HDFS) and queried using Hadoop analytical software called Hive. Sentiment analysis on these comments is conducted using Python. The study assesses the system's effectiveness through self-designed queries on YouTube data with execution times tabulated and graphically presented. The sentiment analysis results affirm the system's capability to discern insightful perspectives from user comments. Nawaz et al. (2019) present an innovative method for evaluating the recommendation effectiveness of YouTube video content through quantitative sentiment analysis of comments and replies. Utilising the Google API the YouTube video comments are extracted in CSV format and sentiment is calculated after essential data preprocessing. Two approaches determine the video's recommendation label, involving the assessment of positive replies and the average sentiment of replies. The normalised score categorises the video into four labels: not recommended, may be recommended, recommended and highly recommended. Trying different YouTube videos shows that the method is really good at improving how videos are chosen for any search on YouTube.

The major issues addressed include enhancing sentiment analysis accuracy, comprehensive analysis of YouTube comments and evaluating recommendation effectiveness. These studies shed light on gaps related to the need for advanced methodologies and systems in sentiment analysis on YouTube. These studies teach us new things about emotions on YouTube. They use clever techniques like machine learning like Hadoop systems and numbers to see how good recommendations are. This study deals with the important issues such as improving how we understand emotions in YouTube comments and checking if recommendations are effective enough. The study shows we need better ways and systems to understand emotions in comments on YouTube.

Future research could look into combining smart computer methods with big data systems to better understand feelings on YouTube. We can also check how well these methods work on other social media sites.

3 Methodology

The Recommendation System employing YouTube Comments Analysis consists of three core components. Firstly, the 'Data Preparation' phase involves collecting and preprocessing YouTube comments with NLP techniques, ensuring effective data division. Subsequently, the 'Model Training' component utilizes supervised learning algorithms like logistic regression or support vector machines to predict user product preferences based on their comments. Finally, the "Model Evaluation and Deployment" phase assesses the model on the testing set and, if successful, deploys it for recommending products to users. The accompanying flow diagram shown in Figure 1 illustrates this process, depicting steps from gathering YouTube links to model deployment.

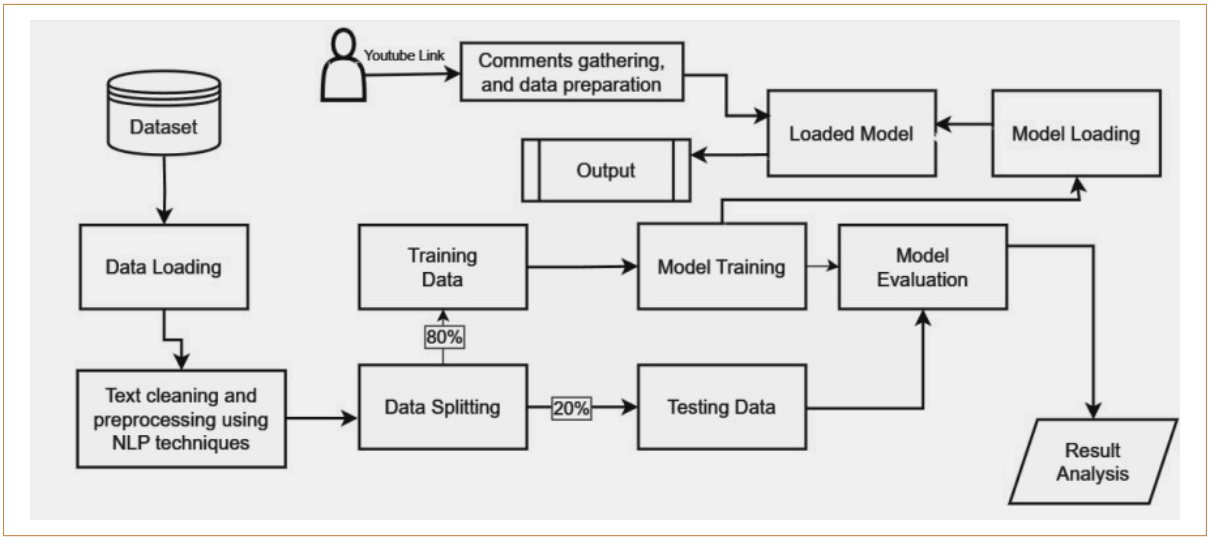


Figure 1: System Architecture

3.1 Data Preparation Process

Data preparation is super important in managing information. It means sorting and splitting data based on certain rules to make things organized, safe, and work better. The main goals are to keep data safe, protect private info, and make it easy to get. For keeping things safe, data preparation helps stop unauthorized people or systems from getting in and causing trouble. It does this by putting data into groups, each with its own rules about who can see and use it. This is like having layers of protection so only the right people or systems can get to the data they need. This stops leaks and stops people from using data without permission. Following rules is a big part of data preparation, especially when dealing with important industries like finance or healthcare. Rules often say data has to be split up to keep privacy safe and make sure it's handled right. Getting data quickly is easier with data preparation. When data is put into groups that make sense, finding, looking at, and getting the right info is faster. This is super important when there's a lot of data because it helps make better decisions faster. So, in short, the data preparation process is a smart way of collecting, cleaning, splitting, and following rules. It makes sure the data used to teach the recommendation system is not just right and important but also follows the rules everyone has to follow.

3.2 Data Cleaning

For the data preprocessing phase the preprocessing textual data is cleaned by removing emoticons, emojis, URLs, Twitter handles and non-alphanumeric characters while handling contractions with the help of a function. Language detection is employed to ensure the text is in English, and non-English text are not considered. The function is applied to a DataFrame column containing text data. The code utilizes regular expressions, the emoji_pattern, and external libraries like contractions and langdetect. The resulting cleaned text is then added as a new column, 'cleaned_text,' in the DataFrame. The Table 1 below demonstrates a comprehensive text cleaning process, addressing various elements, and preparing the data for further analysis or natural language processing tasks.

sentiment	**text**	**cleaned_text**
Negative	@fragilemuse the book is awesome. there are so...	the book is awesome. there are some other grea...
Positive	@LoreleiSpencer I stand corrected! Up market ...	i stand corrected! up market firies....
Positive	sigh. a stupid soppy love story. if only this ...	sigh. a stupid soppy love story. if only this ...
Positive	breakfast! BODY SHOP @ CLUB TROPICANA TONIGHT!...	breakfast! body shop @ club tropicana tonight!...
Positive	@YouLookGreat @libertygrrrl @Sir_Almo free cho...	free chocolate check it out and come play wit...

Table 1: Data Cleaning performed on the tweets

3.3 Data Loading

The data loading phase reads a CSV file is read with the specified encoding and without a header. Subsequently, column names are assigned to the DataFrame, labeling them as 'sentiment', 'id', 'timestamp', 'query', 'user', and 'text'. The displayed DataFrame preview in Figure 2 provides a snapshot of the structure and content of the loaded data for initial exploration and understanding. The figure consists of 6 columns. The first is 'sentiment' which indicates if the text is a negative or a positive sentiment. Next is the id of the tweet, the time when the tweets was posted, the 'query', the user id of the tweet and lastly the text that was posted.

	sentiment	id	timestamp	query	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

Figure 2: Data loading of the tweets

3.4 Histogram for 'sequence_length' feature

The histogram in the Figure 3 below depicts the word count of the tweet that has been tweeted. The higher value is the range of 10-15 words. Finding out the sequence lengths of each tweet is an important part of exploring and preparation of the data. It helps decide the right length for modeling, find any unusual values and grasp how the data is structured. In some machine learning projects, models need sequences of a fixed length. Checking the distribution of sequence lengths helps decide how to deal with sequences of different lengths, like adding extra data or cutting some off. A histogram of sequence lengths is like a check to make sure the data is good. If there are a lot of really short or really long sequences, it might mean there are problems with the data.

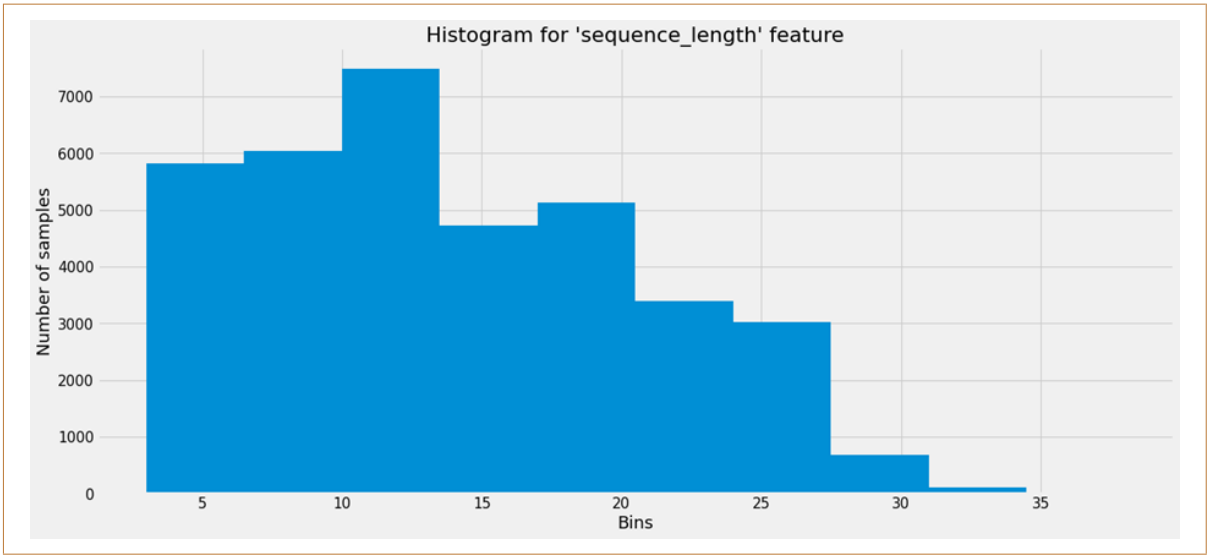


Figure 3: Histogram for the tweets sequence length

3.5 Tools and Test Data

Data analysis and modeling primarily rely on the Python programming language and key libraries such as Pandas, NumPy, Scikit-learn, and Convolutional Long Short-Term Memory (CLSTM) as seen in the Figure below 4. Matplotlib and Seaborn will be utilized for data visualization. The testing dataset will consist of YouTube comments.The code imports necessary libraries, suppresses warnings, and configures display options. It utilizes Pandas, NumPy, Matplotlib and Seaborn for data manipulation and visualization. Scikit-learn functions are employed for resampling, splitting data, and evaluating classification models. The code also imports a custom library "lib_file" from the specified path.

```
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
pd.set_option("display.max_columns", None)
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn.utils import resample
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
from lib_file import lib_path
```

Figure 4: Imported libraries

3.6 Dataset

The dataset is structured as a CSV file, containing six distinct fields. In total, the Sentiment140 dataset comprises a substantial volume of 1,600,000 data points, and it encompasses two primary sentiment targets.[1]

3.7 Data Pre-processing

In the data preprocessing phase the data for a GPT-2 model goes through several steps. First the GPT-2 model and its tokenizer is set up. Next the cleaned text data is broken down from a DataFrame into chunks the model can handle using tokenization. This happens in a loop, turning each cleaned text entry into tokens. The GPT-2 model processes these tokens, giving a final hidden state and convert them into a python list. These lists of features are then put into a new DataFrame with flexible column names. A' new target' column is added that brings in sentiment labels from the original dataset as seen in the Figure 5. The outcome is a readymade dataset with all the extracted features.

26]:

7	column_758	column_759	column_760	column_761	column_762	column_763	column_764	column_765	column_766	column_767	column_768	target
7	0.160584	0.033378	0.120909	-0.145559	-0.073638	3.436818	0.108928	0.127978	-0.028384	0.108947	-0.078783	Negative
0	0.109930	-0.255589	0.009050	0.099625	0.008922	1.151782	-0.086659	-0.017578	-0.120640	0.093889	0.086966	Positive
7	0.156082	0.059024	0.506573	0.178634	0.047518	0.046044	-0.070339	-0.046396	-0.242790	-0.077329	0.186422	Positive
9	-0.053115	-0.101012	0.180936	0.093530	-0.145988	2.738538	0.067402	0.069994	0.075202	0.007824	0.241993	Positive
3	0.032901	0.032167	-0.031477	-0.050620	0.104468	0.641745	0.017091	0.040041	0.067451	-0.002459	0.047438	Negative

Figure 5: Processed data after GPT-2 model

3.8 Analyzing ratings feature

The Figure 6 shows a visual representation of the analysis, focusing on exploring user ratings in different situations. Understanding user sentiments, especially in areas like

[1]URL: https://www.kaggle.com/datasets/kazanova/sentiment140/

product reviews or movie ratings is important for data exploration. The x-axis represents different sentiment categories (e.g., positive, negative) while the y-axis represents the number of samples associated with each sentiment category. This visual representation is in the form of histograms making it easy to see how ratings are spread and spot patterns or outliers. As seen in the Figure the value of positive and negative are equal that 1000. The figure is helpful because it gives a detailed view of how the users feel which can help with businesses to make decisions and to improve products or recommend content.

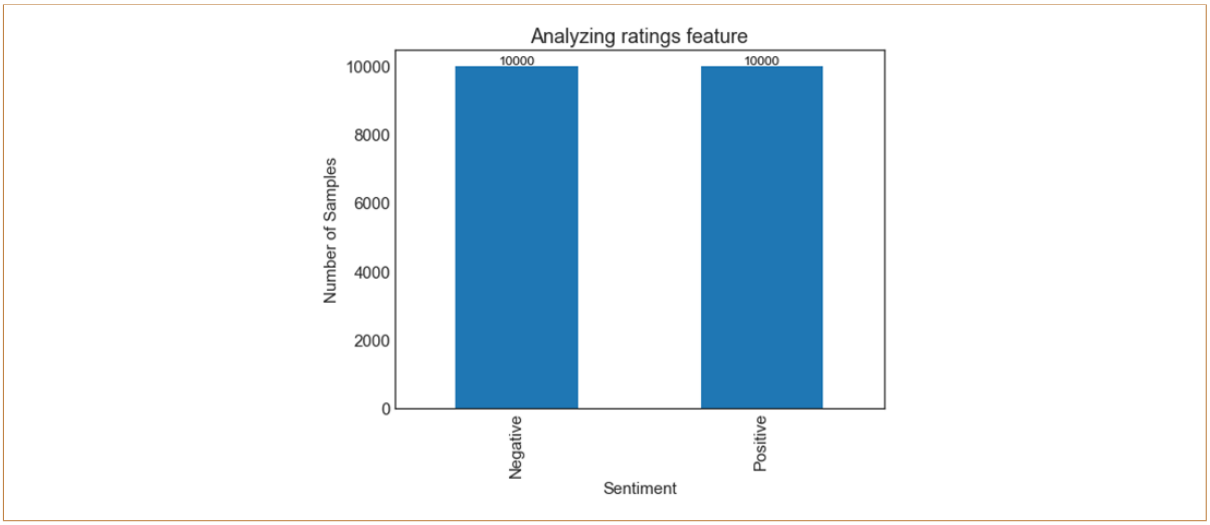


Figure 6: Rating Feature

4 Design Specification

Creating a Convolutional Long Short-Term Memory (CLSTM) model involves designing a deep learning structure specialised for analysing the sequential data. This architecture is used for tasks like sentiment analysis or time series prediction, combines convolutional layers to capture spatial patterns and LSTM layers to capture temporal dependencies, making it effective for analyzing sequences. Input data, such as text sequences or time-series data, undergoes convolutional filtering and LSTM processing to extract hierarchical features and capture long-term dependencies. The number and configuration of layers, filter sizes, and LSTM units should be optimized based on the specific task requirements.

4.1 Convolutional Long Short-Term Memory

Convolutional Long Short-Term Memory networks (CLSTMs) represent a fusion of two powerful neural network architectures: Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs). This combination enables CLSTMs to effectively process grid-like structured data such as images or time series. CNNs are excellent at automatically extracting relevant features from input data. They use filters or kernels to scan the data that capture the spatial hierarchies of features. This makes them well suited for tasks like image recognition. LSTMs, on the other hand, are proficient in modeling sequential or time-dependent patterns. They have a mechanism that helps them choose what information to keep and what to forget over long sequences, making them good at tasks where context and order matter.

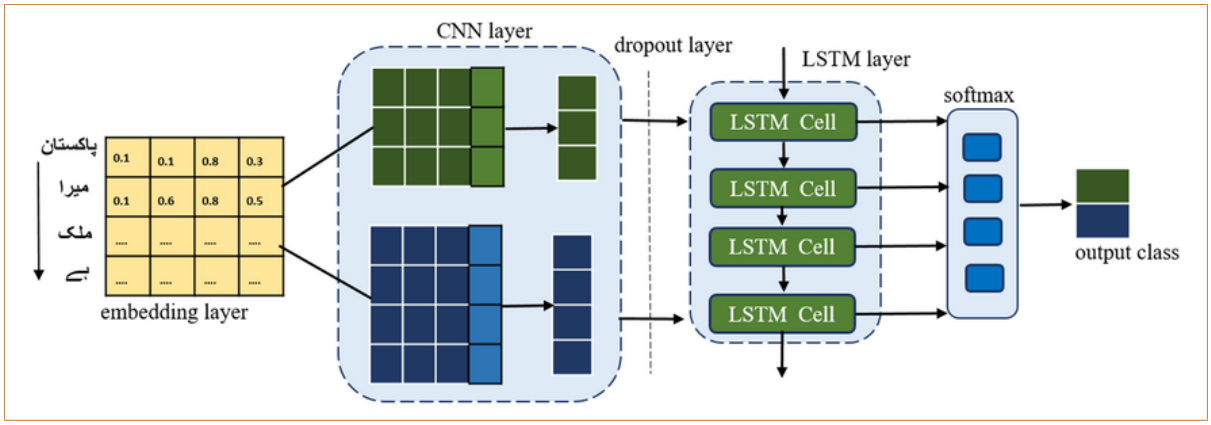


Figure 7: CLSTM algorithm architecture

CLSTMs combine the strengths of Convolutional layers and LSTM units.They can learn spatial features from data which is crucial for tasks like object detection in images. They can grasp extended connections and time-based patterns crucial for tasks requiring understanding of data sequences, such as video analysis or natural language processing.

CLSTMs have found success in various domains including video action recognition, weather forecasting, and medical image analysis. They use CNNs for local information processing and LSTMs for memory retention, ensuring accurate predictions on data with both spatial and sequential features. The architecture of CLSTM is depicted in Figure 7[2].

It's used for tasks that involve sequential data and can capture both spatial and temporal dependencies. To use CLSTM in a mathematical equation, you'd typically express it as a set of equations representing the computations happening in each layer. However, it's important to note that this would be a complex set of equations involving operations like convolutions, matrix multiplications, and activation functions. Here are the equations for a simplified version of a CLSTM:
Let:

- X_t be the input at time t.
- H_t be the hidden state at time t.
- C_t be the cell state at time t.
- W_f, W_i, W_c, W_o be the weight matrices for the forget gate, input gate, cell gate, and output gate respectively.
- U_f, U_i, U_c, U_o be the weight matrices for the forget gate, input gate, cell gate, and output gate respectively (these are the recurrent weights).
- b_f, b_i, b_c, b_o be the bias terms for the forget gate, input gate, cell gate, and output gate respectively.

The equations for a single time step in a CLSTM layer can be written as:

[2]https://www.researchgate.net/figure/The-architecture-of-CLSTM-Model$_fig6_350574964$

1. **Forget Gate (Equation 1):**

$$f_t = \sigma(W_f * X_t + U_f * H_{t-1} + b_f)$$

The forget gate (f_t) determines what information from the previous cell state (C_{t-1}) should be discarded. It uses the sigmoid function (σ) to squish values between 0 and 1. W_f, U_f, and b_f are weight matrices and bias term associated with the forget gate. X_t is the input at the current time step.

2. **Input Gate (Equation 2):**

$$i_t = \sigma(W_i * X_t + U_i * H_{t-1} + b_i)$$

The input gate (i_t) decides what new information should be stored in the cell state. It also uses the sigmoid function. W_i, U_i, and b_i are weight matrices and bias term associated with the input gate.

3. **Cell Gate (Equation 3):**

$$c_t = \tanh(W_c * X_t + U_c * H_{t-1} + b_c)$$

The cell gate (c_t) computes the new candidate values for the cell state. It uses the hyperbolic tangent function (tanh), which squashes values between -1 and 1. W_c, U_c, and b_c are weight matrices and bias term associated with the cell gate.

4. **Update Cell State (Equation 4):**

$$C_t = f_t * C_{t-1} + i_t * c_t$$

The update to the cell state (C_t) is a combination of forgetting old information and adding new information. f_t scales the previous cell state, and $i_t * c_t$ scales and adds the new candidate values.

5. **Output Gate (Equation 5):**

$$o_t = \sigma(W_o * X_t + U_o * H_{t-1} + b_o)$$

The output gate (o_t) determines the next hidden state (H_t). It uses the sigmoid function. W_o, U_o, and b_o are weight matrices and bias term associated with the output gate.

6. **Hidden State (Equation 6):**

$$H_t = o_t * \tanh(C_t)$$

The hidden state (H_t) is the final output of the CLSTM cell. It is computed by applying the output gate to the hyperbolic tangent of the updated cell state.

In summary, these equations govern the flow of information through an LSTM cell in the context of a Convolutional Long Short-Term Memory (CLSTM) layer. They allow the model to selectively retain or discard information, update the cell state, and compute the hidden state for the next time step.

5 Implementation

5.1 Model Loading

The Model loading phase loads a pre-trained Convolutional Gated Recurrent Unit (CGRU) model from a specified file path, initializing it without compilation as seen in Figure 8. Addi- tionally, it incorporates the GPT-2 language model and tokenizer from the Hugging Face library. The GPT-2 model is pre-trained on a large corpus, while the tokenizer enables text processing compatibility. The sentiment-analysis pipeline is also instantiated, offer- ing a simplified interface for sentiment analysis tasks. Lastly, the display.clear output() function is used, possibly to clear any previous outputs. This step prepares a sentiment analysis environment with a pre-trained CGRU model and GPT-2 components for further natural language processing tasks

```
model = load_model("models/ConvolutionalLongShortTermMemory_model.h5", compile=False)
gpt2_model = GPT2Model.from_pretrained('gpt2')
tokenizer = GPT2Tokenizer.from_pretrained('gpt2')
```

Figure 8: Implementaion of GPT-2 Model

5.2 YouTube Video Comments

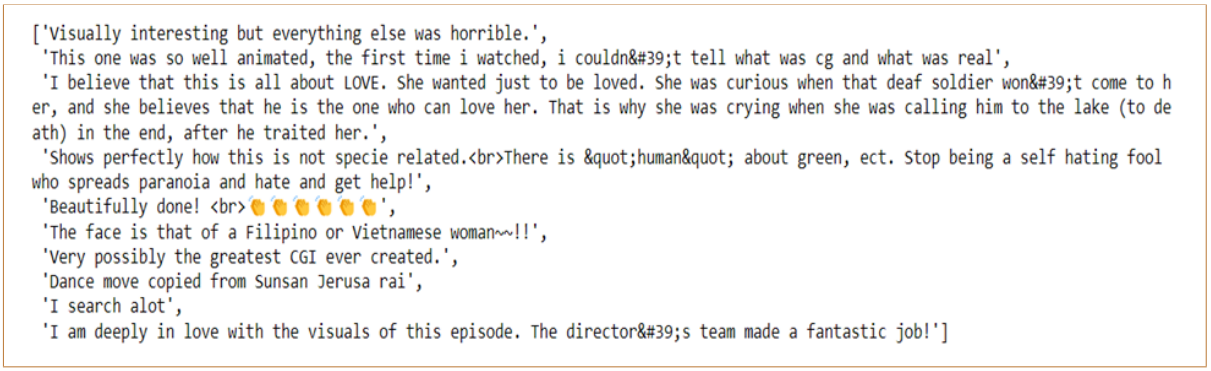
In this phase a user input YouTube video URL[3] is used to extract the video ID from the provided link using the get_video_id_from_url function and checks if a valid video ID is obtained. If successful then it prints the detected video ID; otherwise it prompts the user to try a different URL. Subsequently the code retrieves all comments associated with the given video ID using the retrieve_all_comments function as seen in Figure 9. The retrieved comments are stored in the 'all_comments' variable which excludes the first element and the first ten comments are displayed. This code serves as an initial step in collecting and processing comments from a specified YouTube video for further analysis or natural language processing tasks.

```
['Visually interesting but everything else was horrible.',
 '@pj explain 😊',
 'This one was so well animated, the first time i watched, i couldn&#39;t tell what was cg and what was real',
 'নেটফ্লিক্সে দেখেছি',
 'I believe that this is all about LOVE. She wanted just to be loved. She was curious when that deaf soldier won&#39;t come to h
er, and she believes that he is the one who can love her. That is why she was crying when she was calling him to the lake (to de
ath) in the end, after he traited her.',
 'Shows perfectly how this is not specie related.<br>There is &quot;human&quot; about green, ect. Stop being a self hating fool
who spreads paranoia and hate and get help!',
 'Каждый раз когда Смотрю, понимаю у каждого своя правда жизни😳',
 'Beautifully done! <br>👏👏👏👏👏👏',
 'The face is that of a Filipino or Vietnamese woman~~!!',
 'Very possibly the greatest CGI ever created.']
```

Figure 9: Inputted YouTube Comments

[3]Video URL: https://youtu.be/hrRvFS1qIAQ?si=JPYOy7yh6UhwMQFn

Next step is to iterates through all the comments retrieved from a YouTube video that uses the tqdm library for progress visualisation. For each comment it utilises the is_english_sentence function to determine whether the sentence is in English. If the result is positive, signifying that the comment is in English, it is added to the 'english_comments' list. The process continues for all comments while ensuring that only English-language comments are retained as compared with Figure 9. The Figure 10 below displays the first ten English comments from the filtered list. This step is crucial for refining and isolating English-language comments which smoothing analysis or processing tasks specific to the English subset of comments.



```
['Visually interesting but everything else was horrible.',
 'This one was so well animated, the first time i watched, i couldn&#39;t tell what was cg and what was real',
 'I believe that this is all about LOVE. She wanted just to be loved. She was curious when that deaf soldier won&#39;t come to h
er, and she believes that he is the one who can love her. That is why she was crying when she was calling him to the lake (to de
ath) in the end, after he traited her.',
 'Shows perfectly how this is not specie related.<br>There is &quot;human&quot; about green, ect. Stop being a self hating fool
who spreads paranoia and hate and get help!',
 'Beautifully done! <br>👏👏👏👏👏👏',
 'The face is that of a Filipino or Vietnamese woman~~!!',
 'Very possibly the greatest CGI ever created.',
 'Dance move copied from Sunsan Jerusa rai',
 'I search alot',
 'I am deeply in love with the visuals of this episode. The director&#39;s team made a fantastic job!']
```

Figure 10: YouTube Comments after retaining the English sentences

The final stage is to prints statistics related to sentiment analysis results. The first line from the Figure 11 displays the total number of useful English comments in the DataFrame, which is 444. The next lines provide a breakdown of sentiment categories with 220 comments labeled as 'POSITIVE' and 224 comments as 'NEGATIVE.' These prints offer a concise summary of the sentiment distribution within the dataset, providing insights into the overall sentiment balance and facilitating a quick assessment of the sentiment analysis outcomes.

```
Total useful english comments: 444
Posititive comments: 220
Negative comments: 224
```

Figure 11: results

6 Evaluation

A classification report in data analytics summarises how well a classification model performs. Classification is a type of supervised learning that predicts categorical class labels for new instances using the past observations.

A classification report includes several metrics that help to evaluate the performance of a classification model. The key metrics commonly found in a classification report include:

1. Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positives. It is also called Positive Predictive Value. The formula is given by:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

2. Recall (Sensitivity or True Positive Rate)

Recall is the ratio of correctly predicted positive observations to all observations in the actual class. It is also called True Positive Rate or Sensitivity. The formula is given by:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

3. F1-Score

F1-score is the weighted average of Precision and Recall. It is a good way to show a balance between precision and recall. The formula is given by:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

4. Support

Support is the number of actual occurrences of the class in the specified dataset.

5. Accuracy

Accuracy is the ratio of correctly predicted observations to the total observations. The formula is given by:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Observations}}$$

6. Confusion Matrix

A confusion matrix is a table that describes the performance of a classification algorithm. It is particularly useful for understanding the tdifferent ypes of errors in a model like false positives and false negatives.

6.1 Convolutional Long Short-Term Memory (CLSTM)

6.1.1 Classification Report– CLSTM

The table 2 shows metrics for a binary classification model with 2000 instances each for 'negative' and Positive' classes. Precision values reveal 84% accuracy for predicted negatives and 96% for predicted positives. Recall values show correct identification of 97% of actual negatives and 81% of actual positives. The F1-score, a blend of precision and recall, is 0.90 for "negative" and 0.88 for "positive." The model achieves 89% overall accuracy, accurately classifying instances. Macro and weighted averages for precision,

Class	Precision	Recall	F1-Score	Support
Negative	0.84	0.97	0.90	2000
Positive	0.96	0.81	0.88	2000
Accuracy			0.89	4000
Macro Avg	0.90	0.89	0.89	4000
Weighted Avg	0.90	0.89	0.89	4000

Table 2: Classification Report

recall, and F1-score are consistent at 0.90, indicating stable performance across classes. Despite high precision, there's a recall trade-off, notably for the "positive" class, suggesting room for improvement in capturing all positives. To summarize the model performed very well and the detailed metrics show how effective it is for a binary classification.

6.1.2 Confusion Matrix – CLSTM

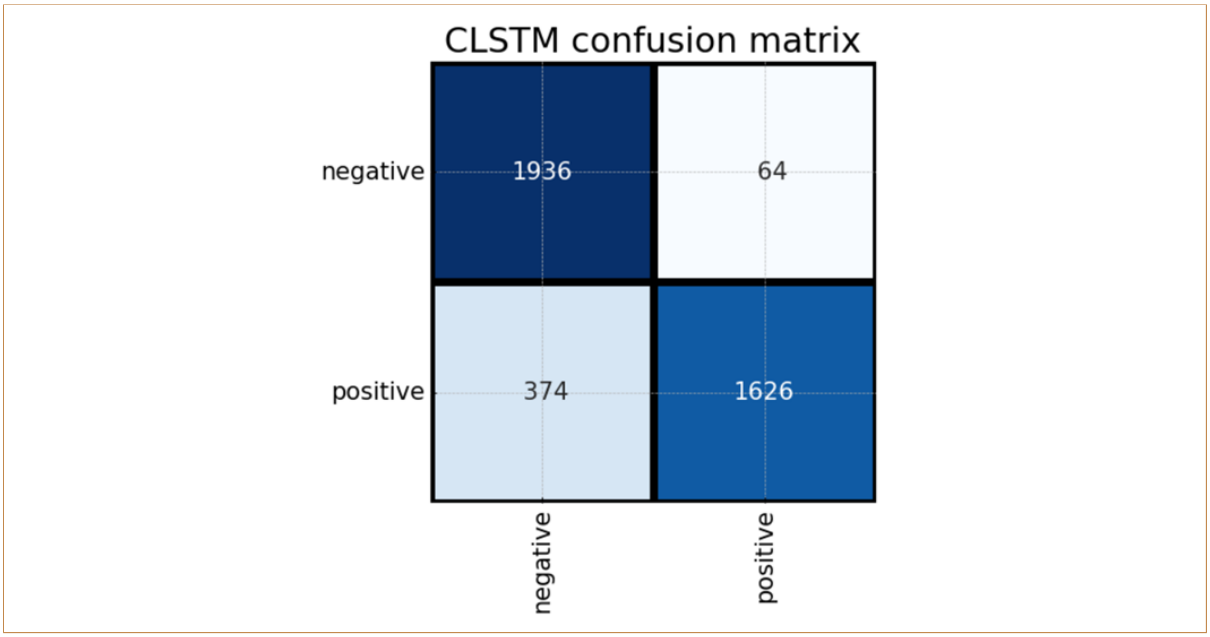


Figure 12: Confusion Matrix

The confusion matrix for the CLSTM model, as depicted in the above Figure 12, illustrates its performance on a dataset comprising a total of 2000 samples. In the negative target class, the model correctly predicted 1936 instances, but it misclassified 64 samples. Conversely, in the positive target class, 1626 predictions were accurate, while 374 were icorrect.

The chart shows how well the model has categorises the samples which not only helps to understand the strengths of the matrix but also its weakness with the scope of improvement. While it's good at identifying negatives, there are misclassifications in both classes, indicating a need for improvement in overall accuracy.

6.1.3 Training Performance Characteristics

1. **Accuracy plot Graph**

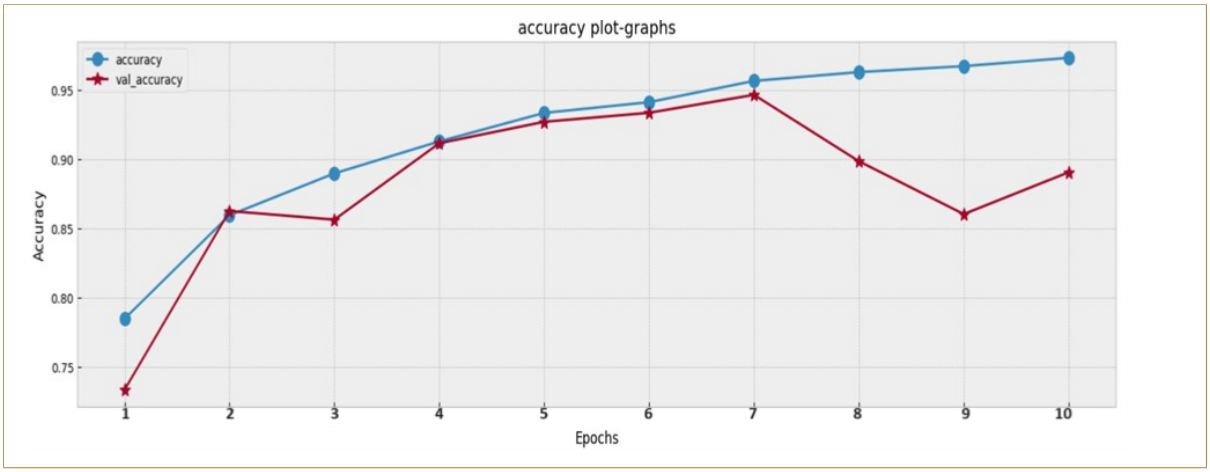


Figure 13: Accuracy plot

An accuracy plot graph visually represents the performance of a machine learning model over epochs. Typically, the x-axis represents epochs or iterations, while the y-axis depicts the corresponding accuracy values. The two y-axis line depicts the training accuracy and validation accuracy of the model over different epochs or iterations. The training accuracy line shows how well the model is learning from the training data as training progresses. The validation accuracy line represents the validation accuracy of the model over epochs. Validation accuracy measures how well the model generalizes to new, unseen data that it hasn't been trained on. It helps assess the model's ability to perform well on data it hasn't encountered during training. A rising trend in accuracy indicates improving model performance, while fluctuations or plateaus may suggest challenges or the need for adjustments. As per the Figure 13 it is clear that the model is performing well as the both the accuarcy lines are rising upwards. But towards the end a small drop and then a rise in the validation accuracy suggest a brief challenge for the model in understanding new validation data during specific epochs. This could be due to complexities faced during training. Since the line by the end bounces back and starts going up again, it might mean the model is resilient, overcoming temporary issues and getting better at generalizing.

2. **Loss plot Graph**

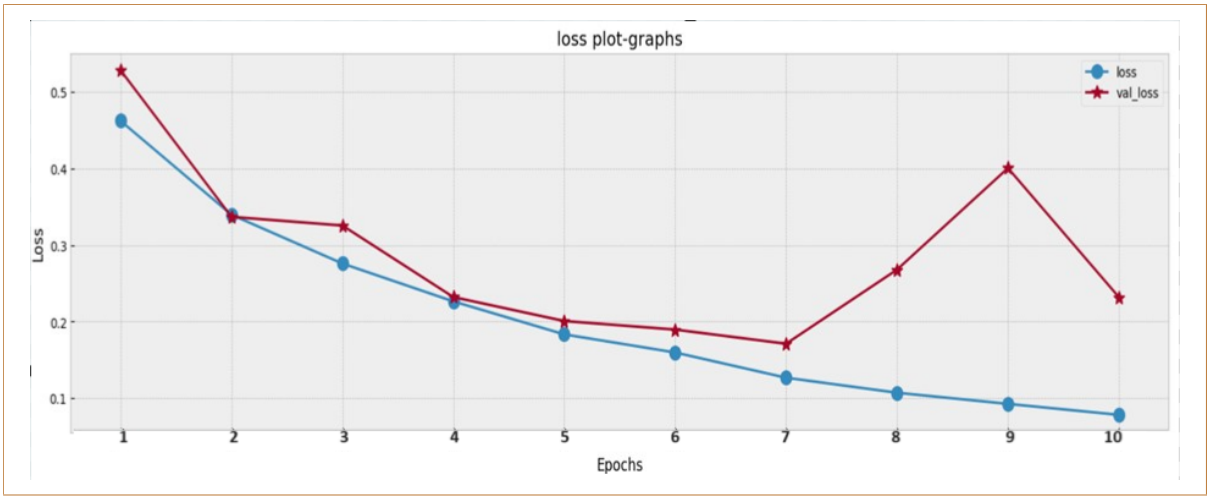


Figure 14: Loss Plot

A loss plot graph is a visual representation of the loss function values of a machine learning model over time or iterations. The x-axis typically represents epochs or iterations, while the y-axis shows the corresponding loss values. the two y-axis line depicts the training loss and validation loss of the model over different epochs or iterations. The training loss line indicates how well the model is minimising errors and fitting the training data. While the validation loss measures how well the model generalizes to new, unseen data. It helps assess how the model is performing on data it hasn't been trained on.By plotting both losses on the same graph helps to know how the model's loss evolves during training and validation.A decrease in both lines as seen in the Figure 14 suggests the model is learning effectively and generalizing well. However, as seen that the training loss continues to decrease while the validation loss increases, it may indicate overfitting, where the model is memorizing the training data but struggling to generalize to new data.

6.2 Discussion

This project can trust the results because the CLSTM algorithm has consistently excelled and been thoroughly validated for its effectiveness in sentiment analysis. The 89% accuracy rate shows that the algorithm is reliable in understanding sentiments from real-time YouTube comments much more than the traditional methods. It's important to note that sentiment analysis benchmarks can vary based on datasets and metrics and future comparisons with additional benchmarks would enhance result validity. The project combined CLSTM based sentiment analysis with an advanced recommendation system tailored for YouTube comments. This new method is much better than the standard search engines because it selecting comments from YouTube videos that are posted by actual people. Analysing the sentiments involves understanding the users' changing preferences and opinions about different products discussed as comments. While effective within the YouTube context the validity of CLSTM across diverse content types and platforms warrants further exploration. As a future research one can investigate how well it can handle different types of data and sources and making sure to used something other than just YouTube as researched by Wadhwani et al. (2022). Finding out how well the model would works in various language and cultural situations would help determine its applicability to a wider range of user created content. The real-time sentiment analysis

using CLSTM is a notable strength which can provide a dynamic and responsive recommendation system. The 89% accuracy attests to CLSTM's efficacy in discerning subtle sentiments within user-generated content. However, relying solely on YouTube comments introduces a limitation as the system's performance hinges on the nature and quality of available comments. The current implementation primarily focuses on English language sentiments and adapting the algorithm for other languages may require further validation. To enhance the system's adaptability, evaluating CLSTM's performance across various social media platforms is crucial. Extending language capabilities to accommodate a more diverse user base and incorporating user feedback for system refinement are key steps. In conclusion, while our work marks significant advancements, recognising both strengths and limitations is essential. The robust CLSTM, coupled with high accuracy, instills confidence in our approach. Addressing discussed limitations and exploring broader applications will contribute to the continuous evolution and generalisability of our innovative AI-enhanced Product Recommendation System.

7 Conclusion and Future Work

This project creates an advanced AI Product Recommendation System that uses the CLSTM algorithm for sentiment analysis on YouTube video comments. We selected CLSTM because it is good at grasping specific sentiments in text. By applying natural language processing the CLSTM algorithm interprets sentiments in YouTube comments which helps to gain a deep understanding of user preferences for various products. Using YouTube comments as a main data source enhances the precision and personalisation of the recommendation system by capturing real-time user sentiments. The CLSTM algorithm is really good as it gets 89% accuracy which shows that it understands the feelings in what users create. This project wants to change how AI systems work by using CLSTM for understanding feelings in the comments by doing things in a new way compared to the usual methods. It embraces the real-time nature of user-generated content which can offer an innovative dataset for training the recommendation system. The integration of sentiment analysis into the recommendation engine marks a departure from conventional methods while prioritising a profound understanding of user sentiments for more accurate and personalised product recommendations. The real-time sentiment analysis of CLSTM outperforms traditional methods which results in a recommendation system with higher accuracy and user satisfaction. By tapping into authentic sentiments in YouTube comments the system provides a more personalised and relevant product recommendation experience by enhancing the user journey. To make the method work better we can look into different areas for future research. First we can check if CLSTM works well on other social media sites will tell us if it can be used in more places than just YouTube. Second, making the sentiment analysis understand different languages will include more people. Lastly, listening to what users say and using their feedback to make the recommendation system better is important for always making it improve.

References

Alhujaili, R. F. and Yafooz, W. M. (2021). Sentiment analysis for youtube videos with user comments: Review, *2021 International Conference on Artificial Intelligence and*

Smart Systems (ICAIS) pp. 814–820.

Alhujaili, R. F. and Yafooz, W. M. (2022). Sentiment analysis for youtube educational videos using machine and deep learning approaches, *2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)*.

Aribowo, A. S. et al. (2021). Cross-domain sentiment analysis model on indonesian youtube comment, *International Journal of Advances in Intelligent Informatics* **7**(1).

Chakravarthi, B. R. (2022). Hope speech detection in youtube comments, *Social Network Analysis and Mining* **12**(1): 75.

Dabas, C., Kaur, P., Gulati, N. and Tilak, M. (2019). Analysis of comments on youtube videos using hadoop, *2019 Fifth International Conference on Image Information Processing (ICIIP)*, Shimla, India, pp. 353–358.

Mulholland, E. et al. (2017). Analysing emotional sentiment in people's youtube channel comments, *Interactivity, Game Creation, Design, Learning, and Innovation: 5th International Conference, ArtsIT 2016, and First International Conference, DLI 2016, Esbjerg, Denmark, May 2–3, 2016, Proceedings*, Vol. 5, Springer International Publishing.

Nawaz, S., Rizwan, M. and Rafiq, M. (2019). Recommendation of effectiveness of youtube video contents by qualitative sentiment analysis of its comments and replies, *Pakistan Journal of Science* **71**(4): 91.

Oksanen, A. et al. (2015). Pro-anorexia and anti-pro-anorexia videos on youtube: Sentiment analysis of user responses, *Journal of medical Internet research* **17**(11): e256.

Pradhan, R. (2021). Extracting sentiments from youtube comments, *2021 Sixth International Conference on Image Information Processing (ICIIP)*, Shimla, India, pp. 1–4.

Singh, R. and Tiwari, A. (2021). Youtube comments sentiment analysis, *Int. J. Sci. Res. Eng. Manage* **5**(5): 1–11.

Tehreem, T. (2021). Sentiment analysis for youtube comments in roman urdu, *arXiv preprint arXiv:2102.10075* .

Wadhwani, S., Richhariya, P. and Soni, A. (2022). Analysis and implementation of sentiment analysis of user youtube comments, (7703).

Yafooz, W. M. and Alhujaili, R. F. (2021). Sentiment analysis for youtube videos with user comments: Review, *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, India, pp. 814–820.