YouTube Comments Sentiment Analysis

Siri Srujana Karnala (N01613397)
Bhavesh Waghela (N01639685)

Date: 10 Dec 2023

Humber College
Machine Learning & Deep Learning - AIGC-5002-0NA

1. Introduction

Background: The project focuses on sentiment analysis for YouTube comments, aiming to understand viewer reactions to content creators' videos.

Problem Statement: Deciphering diverse sentiments in YouTube comments is challenging, hindering effective community engagement and content development for creators.

Objectives: Develop a sentiment analysis model for YouTube comments to enhance content creators' understanding of viewer sentiments and optimize content strategy.

Significance: In the ML context, sentiment analysis is vital for content creators to adapt, engage effectively, and foster positive community interactions.

Impact: Content creators on YouTube will benefit by gaining insights into viewer sentiments, enabling them to tailor content for better engagement, fostering positive community environments and driving channel growth.

2. Methodology

Data Source: The source for the data is https://www.youtube.com/watch?v=bk-nQ7HF6k4&ab_channel=TheDiaryOfACEO the following video from YouTube, and extract the comments form the YouTube API using Google Console API key and googleapiclient library in python to create a request to YouTube and get the response back based on Video ID.

Data Processing:

1. Text Cleaning: Removed special characters and HTML tags to ensure clean text data.
2. Tokenization: Broke down comments into individual words to prepare for analysis.
3. Stop Word Removal: Eliminated common words to focus on sentiment-carrying content.
4. Normalization: Lowercased all words to maintain consistency in text data.

Cleaning ensures quality data; tokenization prepares for analysis; stop word removal focuses on meaningful words, and normalization maintains uniformity.

Model Selection: Chose Artificial Neural Network (ANN) for baseline sentiment analysis, Convolutional Neural Network (CNN) 1D as it works well with text data, and Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM) for sequence dependency analysis.

Training Process: Adjusted parameters for each model to enhance performance. Tailored architecture for each model based on the unique characteristics of ANN, CNN, and RNN-LSTM. Tuning optimized the model performance, and customizing the architectures maximize effectiveness for each model type.

3. Implementation

Code Structure: The following ML pipeline that was followed for performing sentiment analysis. Extract the comments form YouTube API save them to a csv. Load the csv file into a dataframe and check shape, perform text preprocess of the comments in the dataframe and apply Vader Sentiment Analysis on the comments. Check the distribution of the comments as positive, negative, and neutral comments. Remove the columns which has neutral values, assuming that will not be beneficial for decision making for the content creator. Converting it to a Binary Classification Problem. Split the dataset into training and testing for model inputs and then train the models.

Key Code Snippets:

```
In [585]: # Split the dataset into X and y variables.
          # Map the sentiments - positive = 1 and negative = 0 and save to 'y' variable.
          X = data_cleaned['processed_comment']
          y = data_cleaned['sentiment']
          y = np.array(list(map(lambda x: 1 if x=="positive" else 0, y)))

In [586]: y

Out[586]: array([1, 0, 1, ..., 0, 1, 0])
```

Fig 1.0

Converting the sentiments from Vader to binary value that can be easily processed by Neural Networks.

```
: # Create Embedding Matrix having 100 columns
  # Containing 100-dimensional GloVe word embeddings for all words in our corpus.

  embedding_matrix = zeros((vocab_length, 100))
  for word, index in word_tokenizer.word_index.items():
      embedding_vector = embeddings_dictionary.get(word)
      if embedding_vector is not None:
          embedding_matrix[index] = embedding_vector

: embedding_matrix.shape

: (25393, 100)
```

Fig 2.0

The code creates a 100-dimensional embedding matrix using pre-trained GloVe word embeddings for words in the corpus, assigning vectors to corresponding words and initializing the matrix for subsequent use as weights in the Neural Network as input.

Challenges and Solutions: Without using GloVe embeddings, the challenge lied in the need to train embeddings from scratch, which would require large corpus and computing resources. Introducing a GloVe dictionary that has all the words and its corresponding vectors helped into assigning those vectors to the vocabulary in the comments.

4. Results and Discussions:

Model Performance: All the models performed well but the best performance that we received is from RNN-LSTM model with an accuracy of 0.84 and f1 scores for 0 = 0.80 and 1 = 0.86 respectively.

Visualization: Confusion Matrix and Classification Report for Individual Models stating its performance.

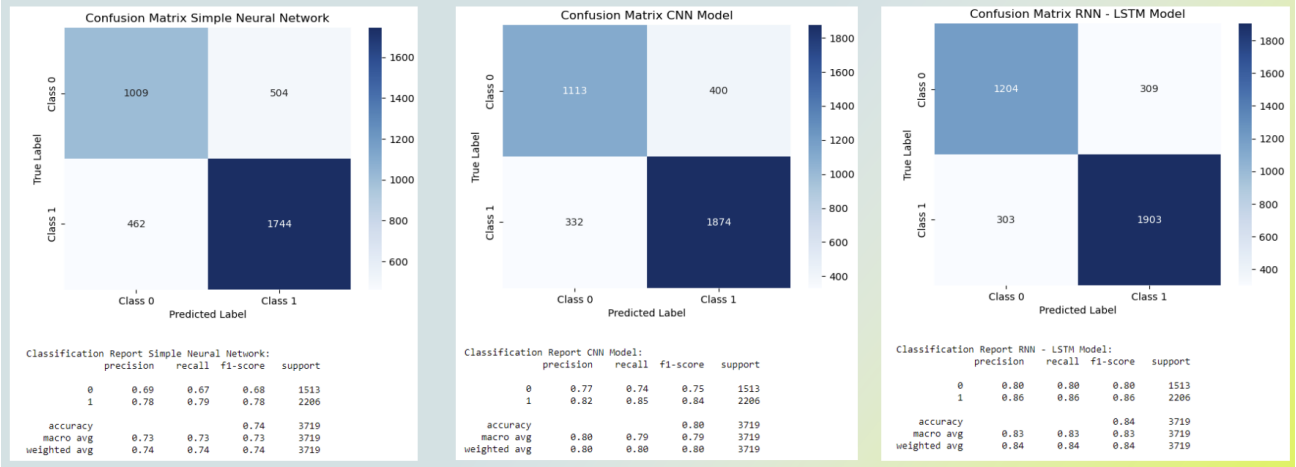
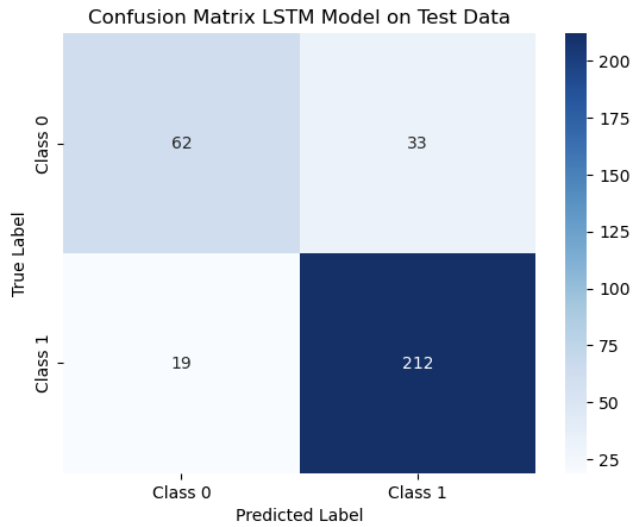


Fig 3.0

The Simple Neural Network achieved an accuracy of 74%, the CNN Model showed improved performance with 80%, and the RNN-LSTM Model exhibited the highest accuracy at 84%. The results suggest that the RNN-LSTM model, leveraging sequential dependencies, outperforms the other models.

Model Evaluation: For evaluation purposes we took another YouTube video https://www.youtube.com/watch?v=6ydFDwv-n8w&t=4s&ab_channel=BloombergOriginals and extracted the comments, applied preprocessing and then sentiment analysis using Vader.

Fig 4.0, Fig 4.1



Classification Report LSTM Model on Test Data:

	precision	recall	f1-score	support
negative	0.77	0.65	0.70	95
positive	0.87	0.92	0.89	231
accuracy			0.84	326
macro avg	0.82	0.79	0.80	326
weighted avg	0.84	0.84	0.84	326

We tried to predict the comment sentiment based on RNN-LSTM model that was trained above and we received an accuracy of 0.84 and the following Classification Report.

5. Ethical Concerns related to YouTube comments sentiment analysis

Data Privacy: The major concern is to safeguard the user privacy in YouTube comments as a right to speech. And to prevent unauthorized access and missues of personal information for other activities without consent of the users.

Bias in Training Data: There may be biases in the training dataset and addressing those biases is a concern as training data may lead to unintended disparities or favoritism towards a particular group of users and that can lead to disparities in the sentiment analysis.

6. Conclusion:

Summary of Findings and limitations: The sentiment analysis project revealed varying accuracies across models—74% for the Simple Neural Network, 80% for the CNN Model, and 84% for the RNN-LSTM Model. Limitations include not including neutral data was not included in the sentiment analysis.

Future Work: Create a more robust model using RNN-LSTM by increasing the number of epochs and hyperparameter tuning. Implement this model with the product review and help e-commerce platforms.

7. Citations:

1. https://www.youtube.com/watch?v=bk-nQ7HF6k4&ab_channel=TheDiaryOfACEO
2. https://www.youtube.com/watch?v=6ydFDwv-n8w&t=4s&ab_channel=BloombergOriginals
3. https://www.youtube.com/watch?v=fklHBWow8vE&ab_channel=StrataScratch
4. https://www.youtube.com/watch?v=zwR6M5zpnWs&ab_channel=SkillC
5. https://github.com/Strata-Scratch/api-youtube/blob/main/README.md
6. https://github.com/hellotinah/youtube_sentiment_analysis/blob/main/cleaned_get_youtube_comments.py
7. https://chat.openai.com/ - For validation purposes, and error solving.