UNITED KINGDOM · CHINA · MALAYSIA

Exploring Sentiment Patterns: A Twitter and Reddit Sentiment Analysis with LSTM and Machine Learning

by JIYA BENNY, MSc Data Science

Submitted to The University of Nottingham
in September 2023
in partial fulfilment of the conditions for the award of the degree of
Master of Data Science

Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This study presents a method for analysing sentiments in social media data from Reddit and Twitter. Understanding viewpoints, trends, and emotions conveyed in online chats is facilitated by sentiment analysis. We compiled a dataset of user-generated content from Twitter and Reddit that covered subjects and feelings in order to undertake our research. To prepare the text data for analysis using LSTM and other Machine Learning models, we cleaned and tokenized the data. On a variety of social or global events, people express their opinions or viewpoints using various media platforms. One of the top social networking services, Twitter and Reddit creates a significant amount of data every day. Their tweets can be categorised using these data depending on the various emotions. To analyse user sentiment, a wide range of technologies are used. Long arrangement data and their complicated dependencies must be handled by a very effective way for sentiment analysis. In this research, we do sentiment analysis on Twitter and Reddit using a deep learning technique and machine Learning Model. For the sentiment analysis, the following techniques are used: Support Vector Machine, Decision Tree, Simple Neural Long Short-Term Memory (LSTM), and Naive Bayes. With the highest accuracy of 83% out of all the suggested strategies, the LSTM is the best. To evaluate the model various metrics like precision, accuracy, recall and F1-score were used. To conduct our experiment, we have gathered a Twitter and Reddit dataset from Kaggle. In order to undertake real-time analytics, the proposed research should be improved in the future to incorporate REST APIs and web crawling-based technologies. For our investigation, around 1.6 million tweets and Reddit data were evaluated.

ACKNOWLEDGEMENT

I want to start by thanking Jonathan Garibaldi, who is my advisor. He taught me how to conduct computer science research, and he continuously encouraged and motivated me as my thesis took shape. He gave me a lot of research ideas and assisted me in writing quality papers.

I want to express my gratitude to my friends and family for their support and patience throughout this academic journey. Their unwavering support has been my pillar of strength.

I am also grateful to my colleagues and research peers at University of Nottingham for stimulating discussions and collaboration. Your contributions enriched my research.

This dissertation would not have been possible without the collective support and encouragement of these individuals and organizations. Thank you from the bottom of my heart.

1.INTRODUCTION

Recent years have seen an exponential increase of data. Data can now be created and gathered very quickly thanks to the growth of the information highway, and it is now so enormous that it has outgrown the capabilities of our traditional applications and processing techniques. Every day, enormous amounts of data are produced in a variety of disciplines, such as those related to travel from automobiles and aeroplanes, financial transactions from the stock market and banking systems, or data we produce in our daily lives. A new era of big data has begun. Big Data includes both diversity and velocity in addition to data volume. It can be data that is both structured and unstructured, in a dumped file or in a high-velocity real-time streaming format. One of the many areas witnessing a data growth is social media. People used to talk to each other nearby back before. Information only circulates within a few selected circles. Nowadays, social networks are used by everyone to speak about everything, including posting messages, uploading photos, and sharing entertaining videos. All of the content on social media platforms is generated by individuals. The term "Internet Minute" has recently been utilised to refer to what happens internet in a minute. In a single Internet Minute, 347,222 tweets were posted to Twitter, 138,889 hours of video were seen on YouTube, and 38,194 photographs were uploaded to Facebook, according to a report by Intel [1]. Our daily lives can no longer function without the social network, and the volume of data generated continues to keep multiplying.

Twitter has become known as one of the most prominent platforms for social networking, platforms for friends to share information and converse. People tweet about a broad spectrum of topics on movies, products, business entities, and many more. Furthermore, Twitter allows users to publish 140 characters in a single tweet, enabling the content straightforward to read and publish. The sentiment of the writers is the most crucial and pertinent piece of data in these tweets. Individual opinions and attitudes concerning the subject of a tweet, such as satisfaction or dissatisfaction, positivity or negativity, were present when the tweet was released. This sentimental information would be an excellent tool for organisations or businesses undertaking consumer surveys and marketing research.

Sentiment analysis can utilise Reddit, which is often referred to as "the front page of the internet," as a broad and interesting source of data. Reddit offers an extensive collection of user-generated material covers an array of concepts, interests, and emotions thanks to its millions of active users and tens of thousands of subreddits. Researchers are able to discover a lot regarding common sentiments, viewpoints, and trends within this enormous online community through conducting sentiment analysis on Reddit data. Reddit data is an unique and informative resource for interpreting sentiment expressions because of the richness of contextual information, interactive style of discussions, confidentiality of users, and subreddit-specific particulars. But it also offers challenges. such as acquiring internet-specific terminology and adapting to various subreddit cultures.

It is generally accepted that emotional analysis has a wide range of applications since persons can have multiple viewpoints on a large range of topics, including politics, religion, the economy, and other topics. This is the reason why as pointed out above, other classifications have been used besides to the both positive and negative ones. It represents an essential technique for collecting and evaluating such data. Indeed, the vast quantity of unprocessed data produced by users of the web in the past few years indicates interest in doing so. To put it another way, it is a way of extracting and categorising sensitive findings in order to get insight about the neutrality and overall attitude portrayed in text through natural language.

For the purpose of sentimental analysis using Twitter and Reddit data to calculate the degree of accuracy, precision (for positive, negative corpuses and neutral), and recall values (for positive, negative corpuses and neutral), method like LSTM and other machine learning models like Support Vector Machine, Decision Tree and multinominal Naïve bayes are implemented. An opinion word that is perceived as favourable may be considered as negative in another context, which is one of the challenges in sentiment analysis. Additionally, the percentage of either positive or negative sentiment has a significant influence on opinions. For instance, one cannot treat "good" and "very good," in the same manner. Despite the fact that a minor alteration to two bits of message does not alter the content of the phrases, this is how standard text processing works. The most recent text mining techniques, however, allow for sophisticated analysis that gauge's word intensity.

1.1MOTIVATION

The capacity to distinguish and harness the feelings expressed by millions of users on sites like Twitter and Reddit has never been more crucial in a society that is becoming more interconnected and where information travels freely across digital platforms. Sentiment analysis, a potent tool for NLP, enables us to decipher the feelings, viewpoints, and mindsets of those who communicate online. Our journey into sentiment analysis reveals a universe of insights that may be used to shape marketing tactics, assess public sentiment, and make decisions in this era of data-driven decision-making. We set out on a mission to derive meaning from the immense sea of social media data by utilising a varied ensemble of machine learning models, including Decision Tree, Naive Bayes, Support Vector Machine, and LSTM. This research serves as a tribute to our dedication to comprehending and using the potential of sentiment analysis in the constantly changing digital ecosystem, where knowledge is more than just a source of power; it also serves as a catalyst for advancement.

1.2 Statement of Research Problem

On platforms like Twitter and Reddit, where millions of users actively share their thoughts and opinions, sentiment analysis can provide valuable insights into public sentiment towards various topics, products, events, or social issues. However, accurate sentiment analysis on these platforms is significantly hampered by the dynamic nature, conciseness, and complexity of user-generated information.

Analysis of sentiment on social media sites like Twitter and Reddit present unique challenges due to the presence of informal language, abbreviations, misspellings, slang, and emoticons in posts. These characteristics make it difficult to accurately extract sentiment from the text. Furthermore, the context-heavy nature of these platforms, where sarcasm, irony, and figurative language are prevalent, can lead to misinterpretation if not properly addressed.

Another significant challenge is the generalizability of sentiment analysis models across different topics and domains. Models trained on one specific topic or domain may struggle to effectively analyse sentiment in other contexts. Therefore, developing techniques that can adapt and transfer knowledge across various domains is crucial for achieving accurate and reliable sentiment analysis.

The research aims to explore and develop a LSTM model that address these challenges and enhance the reliability and accuracy of Twitter and Reddit's sentiment analysis. By doing so, the research can contribute to a better understanding of public sentiment on these platforms, enabling businesses, policymakers, and researchers to make informed decisions based on the analysis of sentiment expressed by users.

2.LITERATURE REVIEW

2.1 Analysis of Related Literature

A subset of data mining to detect the sentiment or polarisation of emotions in subjective content, called sentiment analysis employs a range of methodologies. Studies and corporations have recently begun to pay greater focus to this field of study. Individuals are increasingly looking at how people feel about various topics for a number of reasons. This increase in enthusiasm is partly caused by the social media phenomena that has become a current worldwide phenomenon and its rapidly expanding user base. The following studies are some of the key ones that have made a substantial contribution to the growth and importance of emotional analysis:

APRIL, a unique method for automatic document summarising, is introduced in the paper "APRIL: Interactively Learning to Summarise by Combining Active Preference Learning and Reinforcement Learning," written by Yang Gao, Christian M. Meyer, and Iryna Gurevych[3]. APRIL learns proactively from user choices as opposed to conventional techniques, which rely on referenced descriptions. To lessen the level of complexity associated with current preference-based interactive learning systems, this method combines learning methods including preference learning, active learning, and reinforcement learning. The authors of this paper make a substantial development in the field of selective multi-document summarization by verifying the efficacy of their method through both modelled trials and real-user trials. The report also describes possibilities for the future, such as the investigation of more sophisticated algorithms and the use of APRIL for additional natural language processing tasks.

A Python-based Twitter Sentiment Classification's preliminary processing architecture is provided by a related study conducted by Elias Dritsas, Gerasimos Vonitsanos, and Ioannis E. Livieris in their paper titled "Pre-processing Framework for Twitter Sentiment Classification"[4]. To achieve the most accurate and effective sentiment analysis, this tool is made to handle text and natural language data by removing false positives and noise. The analysis of user-generated content is done in the paper using supervised machine learning algorithms. Quantity of the training sample and different approaches used for selecting features (unigrams, bigrams, and trigrams) are two characteristics that are varied while comparing the efficiency of two classifiers, Naive Bayes and SVM. The primary results of the research and results from experimentation are included in the publication. The study's conclusion involves evaluating performance utilising different datasets through k-fold cross-validation.

The paper "Convolutional Neural Networks for Sentence Classification," published by Yoon Kim[5], discusses various kinds of experiments that used convolutional neural networks (CNN) based on pre-learned vectors of words for language-level classification tasks. According to the findings, even a conventional CNN with minimal hyperparameter modifications and variables that remain constant performs extremely well across a range of rules. The task-specific fine-tuning results in additional enhancement of efficiency. Four out of seven tests, including subject categorization and sentiment analysis, show that these CNN models outperform state-of-the-art techniques. These findings demonstrate the value of pre-trained vectors as feature extractors for a range of datasets. The basic model (CNN-rand) with randomly initialised words performs poorly on its own, whereas the straightforward model (CNN-static) with static vectors outperforms more complex deep learning models that require sophisticated pooling techniques or precomputed parse trees. By fine-tuning the pre-trained vectors for each job (CNN-non-static), performance is increased even more.

The GloVe global logbilinear regression model, developed for learning word vector space representations, is described in the publication "GloVe: Global Vectors for Word Representation"[6]. By concentrating on training solely with nonzero entries in a word-word overlap matrix, this model optimally utilises mathematical knowledge and incorporates the benefits of both global factorization of matrix and local window strategies. As a result, a vector space with significant structure emerges that outperforms comparable simulations for tasks like identified entity recognition and similarity evaluation. The results show that the GloVe model beats alternative initial models regularly, frequently requiring fewer vector dimensions and less extensive texts. Moreover, the GloVe model performs significantly better after being simply trained on a huge 42 billion units dataset. As was first shown with neural vectors in a previous study (Turian et al., 2010), these GloVe vectors show promise for subsequent natural language processing tasks.

Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuanjing Huang introduce an innovative approach known as Cached Long Short-Term Memory neural network (CLSTM) for carrying out sentiment classification at the article level in their investigation study titled "Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification"[7]. The CLSTM uses a technique for caching that separates memories into groups with varying rates of forgetting enabling it to fully capture the semantic content of long texts. On three datasets that are freely accessible used for document-level sentiment analysis, the model they suggest performs better than the efficiency of the most recent state-of-the-art algorithms. In order to deal with concerns with memory constraints, this work specifically addresses the difficulty of modelling lengthy texts in document-level categorization of sentiment within a recurrent architecture. The actual results support the usefulness of the suggested model for document-level text sentiment analysis using an RNN architecture.

A novel hierarchical focus network for document classification is introduced in the thorough study of the Carnegie Mellon University academic team, "Hierarchical Attention Networks for Document Classification"[8]. The two different levels of attention processes included in this model function both at the word- and sentence-levels. Surprisingly, regardless of the amount of information or problem type, this suggested methodology continuously beats current techniques by a wide margin across six substantial text categorization tasks. In comparison to HN-AVE, which primarily depends on uninformative global word and phrase context vectors, the HN-ATT model regularly performs better. The thorough findings from the experiments for each dataset indicate the model's capacity to pick out specific phrases and clauses that provide value to the classification process. This is further demonstrated by the attention layer visualisation.

In a significant study by Robin Jia and Percy Liang titled "Adversarial Examples for Evaluating Reading Comprehension Systems"[9], the researchers present a ground-breaking adversarial assessment approach for the Stanford Question Answering Dataset (SQuAD), which is intended to evaluate the comprehension of language abilities of literature systems. This innovative approach evaluates whether these algorithms can accurately respond to inquiries about passages that contain antagonistic introduced sections. These antagonistic statements are created artificially to confuse computers without changing the right answers or deceiving people who are reading them. The study's conclusions show that when sixteen previously published systems are subjected to an adversarial examination, their accuracy significantly declines, falling from an average F1 score of 75% to 36%. The mean performance of the four models further drops to just 7% when the adversary is allowed to insert grammatical word sequences. The primary goal of the studies is to offer light on the difficulties faced by instances of conflict in comprehension of reading systems and to motivate the creation of more accurate understanding models.

Pang and Lee use a variety of datasets to introduce various methodologies and issues in the field of sentiment analysis. They additionally investigate the effects of phrase frequency and Ngam feature selection on the results. Additionally, their system framework divides posts into n-grams and takes into account unique situations. In particular, they explore the use of sentiment analysis at various levels of hierarchy. Pang et al. evaluate the efficacy of document-level sentiment analysis in a different study, concentrating in particular on multiple highly reviewed films from IMDB. They use a pre-processing framework to categorise Twitter sentiment. Turney et al. choose terms of adjectives, adverbs, and other language elements suggestive of emotional content when evaluating the sentiment of a piece of writing based on its suggested terms. Their study uses user feedback on a range of subjects, including films, destinations for travel, financial services, and automobiles. Wilson et al. examine the MPQA corpus's opinions along with a dataset of articles from journals from diverse sources that have had their psychological content evaluated. As a way to ascertain the overall mood of the text, they first divide terms that are neutral into polar and objective groups before doing polarity research on subjective phrases.

The emotion expressed by customers towards particular attributes and the products was then determined by analysing product reviews. Based on visitor reports of the places they went to; scientists created an illustration for emotional identification. Similar to this, a system was developed to prediction of the results of the American presidential election in 2012 through examining the emotion of tweets posted in social platform by users of Twitter. The authors from the works previously mentioned provided several techniques for automatically analysing tweets and determining each tweet's sentiment using the Ekman emotion model. One or more of the six basic emotions that individuals experience—anger, discontent, terror, happiness, sorrow, and surprise—must exist, according to the Ekman emotion model. Last but not least, "Large scale implementations for twitter sentiment classification"[10] presents a completely new distributed architecture built on Hadoop as well as Spark that leveraging the use of hashtags and emoji within a single tweet. Bloom filters are additionally employed to boost the effectiveness in the suggested method.

2.2. BACKGROUND OF STUDY

This chapter discuss about the background of text categorisation, sentiment analysis and the various algorithms for the procedure.

2.2.1 Text Classification

Text classification is a component of data mining to categorise the content related to the information they contain. News, writings, and literature may all be divided into several categories based on the characteristics we identified.

2.2.2 Process of Text Classification

Considering the text classification, we have to consider the main two steps in supervised classification: the training phase and the testing phase. The tagged corpora dataset creation, training of the classifier, text vectorization, and preliminary processing of the training data are typically included in the training step. Preprocessing, vectorization, and classifying the testing text are all part of the testing stage. Figure 1 shows this mechanism in action.

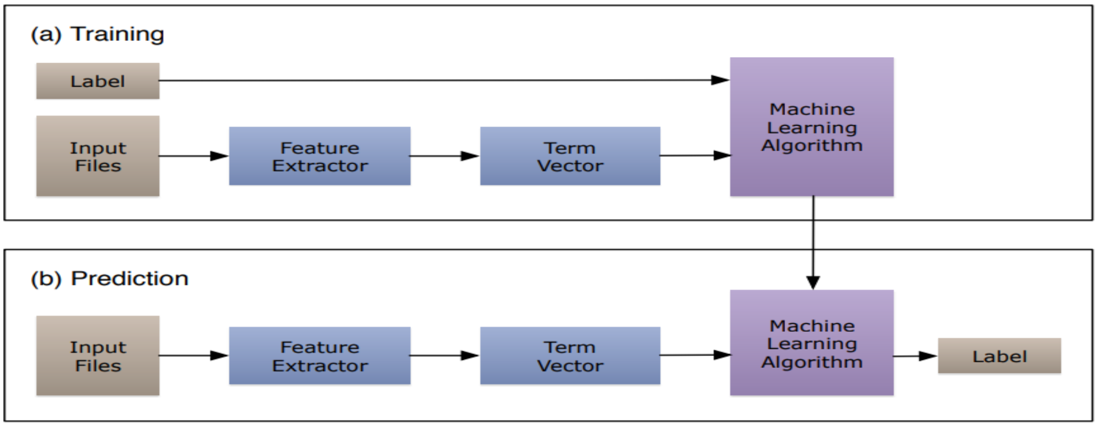


Figure 1 : Supervised Text Classifications

1.Creating Corpus:

Grouping of the text according to their categories. Every text in the dataset has been correctly identified and assigned to a specific category where it belongs. This has been split into training and testing set for further analysis and modelling.

2. Pre-Processing:

Eliminate all superfluous content, including stop words, extra punctuation, and unclear text. This action is crucial since it will have an impact on how the classifier is trained.

3.Vectorization of Text

Create a computer-recognizable vector out of the text. Based on the features we selected all text will be interpreted as vectors with features.

4. Training of the Model:

To build a training model, select one of the text classification techniques from machine learning algorithms or from deep learning techniques and input the training sample to the selected algorithm.

5. Classification

After we train the model using training sample, enter testing data into it to get the prediction classification.

2.3. Sentimental Analysis

Sentiment analysis is a technique for data mining that identifies an author's or the writer's viewpoint regarding a topic or the overall sentiment conveyed within a written work. During the recent years, there are several different sentiment analysis techniques available. To categorise text into a positive or negative emotion class based just on text categorization is the simplest method. The fundamental approach is lexicon-based [11], which involves looking at tweets in terms of the words they include. The texts are scanned to see if any particular emotive terms are present. Some phrases are classified in a dictionary as positive and others as negative, and each is given an emotion score. The score will be used to decide the entire text. Maintaining a lexicon of important phrases to determine the level of sentiment is challenging, though. Because of this, supervised and unsupervised Classification of text algorithms, such as Maximum

Entropy, Support Vector Machines, K-Nearest Neighbours, Naive Bayes, and Decision Trees and various deep learning techniques have also been created and put to use [12]. For these machine learning approaches, the classifier needs to be provided with enough labelled data to train it before it can do the classification.

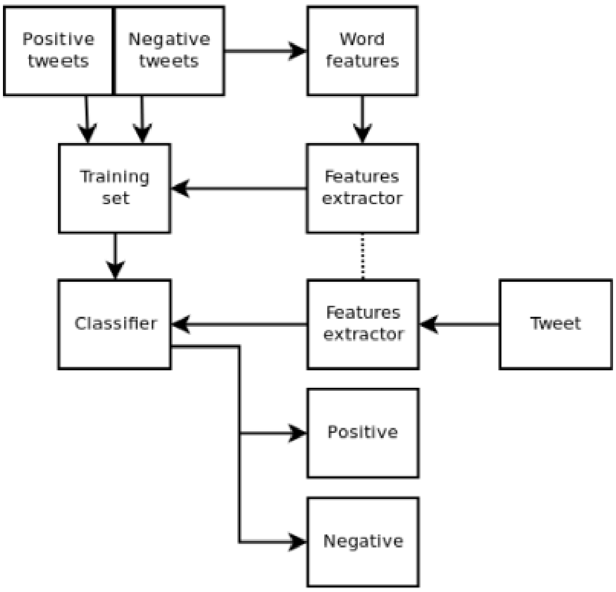


Figure 2: Sentimental Analysis Architecture

2.3.1 Approaches For sentimental Analysis

For the data from Twitter and Reddit, there are primarily two methodologies for sentiment analysis:

- **Machine Learning Approaches**

Text is categorised using a classification methodology and a machine learning-based technique. Two sorts of machine learning approaches are typically used.

a) **Unsupervised learning:** This kind of machine learning method depends on clustering since it doesn't provide the right targets at all and lacks a category.

b) **Supervised learning:** A method is built up on a labelled dataset, and the model is given the labels as it goes along. When utilised in decision-making processes, these datasets with labels have been trained to give findings that are helpful.

The decision to choose and retrieval of the particular collection of attributes used for recognising sentiment plays a key role when evaluating the efficiency of both of these learning methodologies. The machine learning technique that can be used for evaluation of sentiment primarily falls within the category of supervised classification. Two sets of data are required for a machine learning technique:

1) Training data 2) Testing data

The tweets and Reddit data have been categorised using a variety of machine learning techniques. Machine learning techniques like Maximum Entropy (ME), Support Vector Machines (SVM) and Naïve Bayes have been very successful in sentiment analysis. The initial stage in machine learning is collecting training datasets. The training data is then used to train the classifier. A key decision after selecting a supervised classification algorithm is selecting an attribute. They can describe the presentation of documents to us. The characteristics used in sentimental classification most usually are:

- Frequency of terms and its presence
- Information on parts of speech
- Negative Words
- Phrases and words that convey opinions.

The most popular supervised algorithms include Decision Tree, Naive Bayes, and Support Vector Machines (SVM). When it is not possible to have an initial collection of tagged documents or views to categorise the remaining things, unsupervised and semi-supervised procedures are advised.

- **Deep Learning Techniques**

Sentiment analysis, a vital part of natural language processing (NLP), has been transformed by deep learning-based technologies. By utilising neural networks to recognise intricate linguistic patterns, deep learning has dramatically increased the efficiency and accuracy of sentiment analysis. Here is a summary of how sentiment analysis uses deep learning:

a) **Deep Learning:** Deep learning methods, especially neural networks, have shown to be quite successful at sentiment analysis. Popular options such as Convolutional Neural Networks (CNNs), Recurrent

Neural Networks (RNNs), and more recently, Transformers. These models can automatically discover and extract pertinent elements from text, giving them the ability to comprehend linguistic complexity and context.

b) **Recurrent Neural Network:** Recurrent neural networks are excellent for sequential data, such as text. By keeping hidden states that change when new words are analysed, they may identify word dependency. They are less successful at capturing long-range dependencies and have vanishing gradient issues (The network learns poorly or not at all in the initial layers due to the extremely small gradients during backpropagation, of the loss function with regard to the network's parameters.)

c) **Long Short-Term Memory (LSTM):** An RNN variant called LSTMs was created to address the vanishing gradient issue. By identifying both immediate and long-term dependencies in text data, they have demonstrated effectiveness in sentiment analysis.

d) **Convolutional Neural Networks (CNNs):** Originally developed for the processing of images, CNNs may be used for analysing text by treating it as a 2D grid made up of words and their respective positions. They are excellent for tasks like sentiment analysis at the sentence or phrase level since they are good at finding local patterns in text.

e) **Transformers:** Transformers, especially BERT (Bidirectional Encoder Representations from Transformers) models, have raised the standard for sentiment analysis and other NLP tasks. Due to their extensive pre-training on text corpora of words BERT-based models are very good at capturing meaning and grammar. On sentiment analysis tasks, these models can be adjusted for exceptional accuracy.

Lexicon based approaches

Lexicon-based approaches [17] establish the polarity by comparing the data with sentiment phrases from a sentiment vocabulary. They assign a sentiment score to each word in the dictionary, indicating how favourable, Unfavourable and Objective the term is. A sentiment lexicon, which is a database of well-known and precompiled sentiment expressions, phrases, and even idioms made for standard communication genres, such as the Opinion Finder lexicon, is a necessary component of lexicon-based approaches;

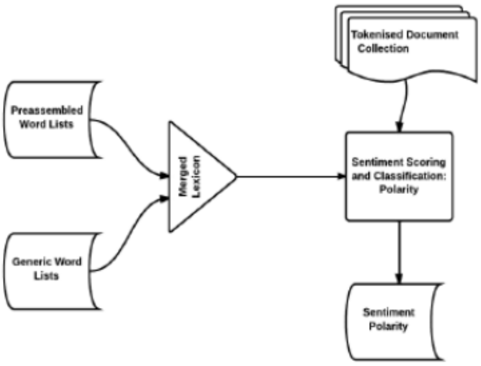

Tokenised Document Collection
Preassembled Word Lists
Merged Lexicon
Sentiment Scoring and Classification: Polarity
Generic Word Lists
Sentiment Polarity

Figure 3:Lexicon based model

1) **Dictionary Based:** It is based on the usage of words (seeds), which are frequently collected and tagged by hand. This set grows by looking up terms in a dictionary's equivalents and contrasting words. That dictionary's example, WordNet, is utilised to build SentiWordNet, a thesaurus. One of the main drawbacks is it can't deal with orientations peculiar to a certain domain or situation.

2) **Corpus-Based:** The corpus-based approach tries to provide lexicon files related to a certain academic discipline. These dictionaries are produced by turning a collection of seed phrases opinion representing words into related ideas, either statistically or lexically.
    - Latent Semantic Analysis (LSA), a statistical technique.
    - Semantic approaches, including the utilising relations from thesauruses like WordNet, or synonyms and antonyms, may also be a useful choice.

Based on performance criteria like recall and precision, we give an unbiased evaluation of the current technologies for opinion mining, including machine learning, lexicon-based approaches, cross-domain and cross-lingual approaches, etc.

2.3.2 Sentimental Analysis Task

Semantic approaches, including the Sentiment analysis is a challenging interdisciplinary subject that incorporates web machine learning, web mining and natural language processing. It is a challenging assignment that can be divided into the following smaller tasks:

- Subjective Classification
- Sentiment Classification
- Complimentary Tasks
  - Object holder Extraction
  - Feature Extraction

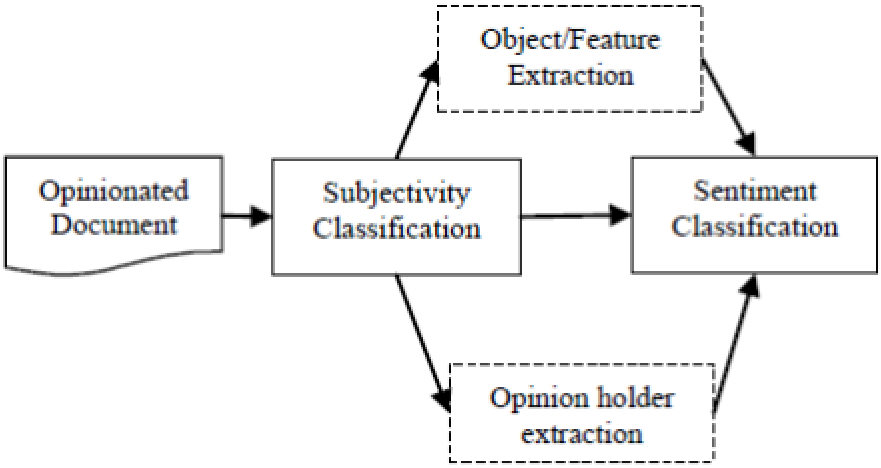

Object/Feature Extraction
Opinionated Document
Subjectivity Classification
Sentiment Classification
Opinion holder extraction

Figure 4: *Sentimental Analysis Task*

[1] **Subjectivity Classification:** The task of classifying sentences as subjective or non-subjective is known as subjectivity classification. Let S = {s1, s2..., s_n } be a group of statements from document D. The challenge of classifying subjectivity is to differentiating between sentences that express opinions and other types of subjectivity (subjective sentences set s_S)and those that offer factual data objectively (objective sentences set s_o), where s_sUs_o=S.

[2] **Sentiment Classification:** Semantic approaches, including the once we've established whether a sentence is opinionated, we need to establish its polarity, or whether it expresses a favourable or unfavourable opinion. Regression, ranking, a multi-class classification (remarkably negative,

negative, neutral, positive, or extremely positive), or a binary categorization (positive or negative) can all be used to classify emotions. Depending on how sentiment analysis is used, the subtasks of opinion holder extraction and object feature extraction may be viewed as optional.

[3] **Complementary Tasks:** Extracting Opinion Holders locating sources or experts who can provide their opinions. Semantic approaches, including the Opinion holder detection refers to the process of locating either indirect or direct sources of opinion.

- Feature/Object Extraction: It is the identification of the intended object.

2.3.3 Levels of Sentimental Analysis

The previous section's tasks can be carried out at various levels of granularity.

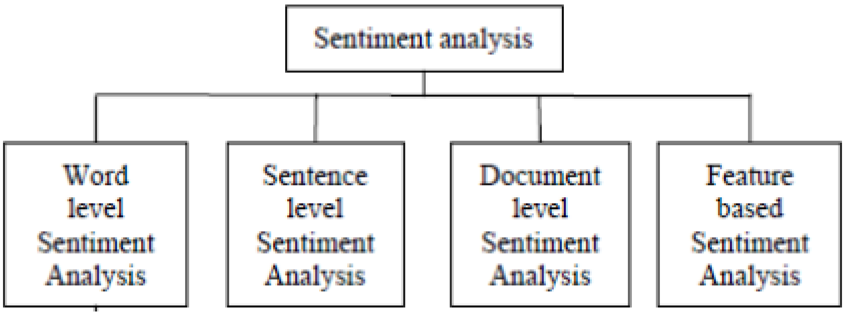

Sentiment analysis
Word level Sentiment Analysis
Sentence level Sentiment Analysis
Document level Sentiment Analysis
Feature based Sentiment Analysis

Figure 5: Levels of Sentimental Analysis

a) Document Level

It has to do with assigning a feeling to each unique document. At the document level, the entire document is classified as either having a positive or negative tone.

General Strategy: To determine the document's polarity, identify the sentiment polarities of certain sentences or words and then combine them. Other strategies include pragmatics, co-reference resolution, and other complex linguistic issues. There are numerous steps involved in it, including:

- Task: Classifying the full document's emotions
- Classes: Positive, neutral and negative

- Assumption: Semantic approaches, including the Each document focuses on a certain topic and expresses the viewpoint of a single perspective holder (unlike conversations in blogs, posts, etc.).

b) Sentence Level

Sentence-level Sentiment analysis involves assigning each sentence its appropriate sentiment polarity. Sentence level classification for sentiment divides sentences into classes that are either good, negative, or neutral. The general strategy is to classify each word's emotional orientation inside the sentence or phrase, then use those findings to estimate the sentiment of the entire composition. Other strategies Considering the text's discourse structure Many tasks are involved in this, including:

- Task 1: Understanding the Difference Between Objective and Subjective Sentences.
- Task 2: Sentence Classification: Positive or Negative and Neutral Sentences
- Assumption: A sentence only contains a single viewpoint, which is not guaranteed to be truthful.

c) Aspect or Feature Level

It deals with giving each sentence an emotion and identifying the thing that the sensation is directed at. The issue of retrieving and identifying product features from the source data is addressed by aspect or feature level sentiment categorization. Discourse structures and dependency parsers are employed in this. Many tasks are involved in this, including:

- Task 1: Determine and extract object characteristics that have been discussed by a person with authority (such as a reviewer).
- Task 2: Determining if comments are neutral, positive, or negative about a feature.
- Task 3: Look up synonyms for feature

d)Word level

Most of the recent publications have used word and phrase polarity prior to grouping sentiment at the sentence and document levels. Adverbs are used less frequently in this process than adjectives, and there are two ways to automatically label emotions at the word level.

1) Dictionary-Based Techniques (2) Approaches based on corpora.

2.4 Application of Sentimental Analysis

1. Applications that utilise website reviews:

Today, there is a huge variety of opinions and feedback available on the Internet on almost anything. This covers, among other things, reviews of products, opinions on services, and political issues. As a result, sentiment analysis requires a mechanism for extracting sentiments related to a certain commodity or service. It will give us the ability to instantly offer comments or reviews for the chosen good, item, etc. This would satisfy the needs of both users and vendors.

2. Applications as a Member Technology:

A sentiment forecasting system can be useful for recommender systems as well. The recommender system will not make recommendations for items that have a significant number of negative comments or poor ratings. In internet conversations, we come across poor language and other elements. These can be quickly identified by recognising a strongly negative attitude and correctly handling it.

3. Application in Business Intelligence

Consumers now prefer to check product reviews online before making a purchase, it has been observed. And for many businesses, online reviews determine whether a product succeeds or fails. Thus, sentiment analysis has a big impact on businesses. Businesses also seek to use the sentiment expressed in online reviews to improve their products because doing so will improve their reputation and make customers happier.

4.Applications across Domains

Sentiment analysis, a technique that shows trends in human emotions, particularly on social media, has recently been used in searches in sociology and other sectors like sports and medicine.

5.Smart home applications

Smart homes are considered to be the newest technology. Future homes will be networked, and residents will be able to control every aspect of the house using a tablet computer. Recent years have seen a lot of research into the Internet of Things (IoT). Additionally, IoT would make use of sentiment analysis. The ambiance of the home, for instance, may be altered based on the user's current attitude or feeling to create a quiet and peaceful atmosphere. Additionally, sentiment analysis can be useful for trend predictions. Monitoring public opinion can provide useful insights into sales trends and consumer satisfaction.

2.5 Famous Algorithms Used

a) Naïve Bayes

One of the most popular supervised classification techniques that can be applied to text classification is naive Bayes [13, 14]. Prior to that, it is important to understand what exactly a feature vector is. We need to choose features from the collected information before conducting labelling. The feature vector, also known as the term vector in text classification, is a highly crucial component throughout the learning and classifying process. All tweet and reddit texts will be converted into term vectors for the classifier to process. Typically, a term vector is created using a singular vocabulary that is drawn from the training sample and contains no redundant words. The vocabulary's size is determined by the term vector. Naive Bayes is implemented in two different ways: Naive Bayes Bernoulli and Naive Bayes Multinomial. The method used for extracting characteristics from the documents is the fundamental distinction between both. Let's use a tweet as a scenario for the Bernoulli approach. A term vectors will be set up with all entries set to zero. The next step is to see if every word in the lexicon is used in the tweet. Mark the corresponding entry in the term vector to 1 if it exists; otherwise, mark it to 0 if it does not. If the vocabulary is large enough, each tweet might be represented using a term vector made up of 0s and 1s.

The term vectors are used in Multinomial Naive Bayes to compute probability for various classes (such as sentiment classes like positive and negative). We use the probabilities to assign the tweet to one of the groups. The probabilities are based on the frequency of words in each class in the training data. The naive Bayes classification system might classify this tweet as "positive" based on the term vector, for instance, if we have two classes, "positive" and "negative," and our training data indicates that the word "happy" is more associated with the "positive" class, while the word "sad" is more associated with the "negative" class.

In text classification, we typically disregard the document's arrangement of words. Instead, we merely take into account whether a particular word is included or not, such as if a term from the lexicon is used in the document or not. A bag of words is the name given to this model. It's as if all the words were thrown into a sack and could appear in any arrangement. The term vector's element can represent the occurrence of a word in addition to whether it appears or missing. Each term in a vector can be imagined as a point with n dimensions in a system of coordinates with n dimensions, where n is the vocabulary size. The data used for training can be viewed as several classes of a collection of n-dimension elements. The text classification problem then changes to a typical problem of classifying points, even though the dimension may be quite huge.

For instance, suppose we have a document A and a class B that contains certain classes. The posterior probability P(B|A) must then be computed, and the biggest value must be selected in order to determine which class the document A belongs to. Bayes' Theorem allows for the computation of P(B|A):

$$P(B|A) = \frac{P(A|B)\,P(B)}{P(A)} \sim P(A|B)\,P(B)$$

where the labelled dataset can be used to calculate the prior probability and likelihood.

These two algorithms operate as follows. Since each class has an equal likelihood, we can readily calculate the prior probability P (B_i). Let vocabulary be called as X and j-th word in the vocabulary X be w_j. So, considering the training set of data, we can calculate the likelihood that a given value ($w_j|C_i$.) belongs to a certain class called C_i.

Consequently, let's assume that S_i is the i-th tweet in my test dataset and T_i is its term vector. As previously mentioned, T_i , comprises 0s and 1s that indicate if a matching word from the vocabulary is present in the tweet S_i. Consequently, the likelihood that a tweet belongs to a class is C_i.

$$P(S_i|C_i) \propto P(T_j|C_i) = \prod_{j=1}^{|V|}[T_j P(w_j|C_i) + (1 - T_j)(1 - P(w_j|C_i))]$$

The formula, which represents the multiple of the odds that this tweet is made up of terms from vocabulary, is simple to understand.

b) Support Vector Machine

One of the well-known supervised machine learning techniques for text classification is the support vector machine (SVM) [14, 15]. Finding a linear separator in the search space that can best separate the various classes is the basic goal of a support vector machine. There are two groups, A and B, as seen in Figure 2. They are divided into two classes by the I, II, and III hyperplanes that connect them. The best separator will be the hyperplane in the accompanying figure, hyperplane I, which has the longest normal distance of all of the data points.

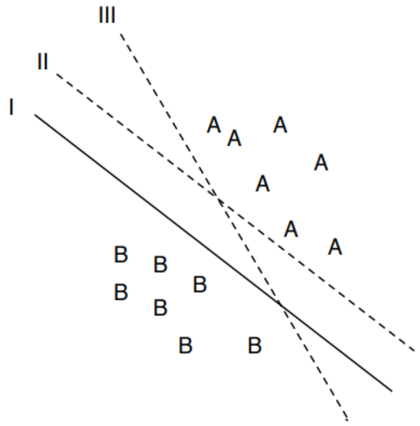


Figure 6: Support vector Machine

c) Decision Tree

A straightforward flowchart that identifies labels for the inputs is commonly referred to as a decision tree [14,16]. The decision nodes in this flowchart verify the attribute values, while the leaf nodes provide labels. To choose the label for an input value, we start at the flowchart's root node, the first decision node. This node has a state that evaluates one of the input value's features and selects a branch based on the feature's value. We follow the branch that describes our input value and arrive at a new decision node with a new condition based on the characteristics of the input value. We follow the branch determined by the conditions of each node until we get to a leaf node that gives the input value a label.

d)Long Short-Term Memory Networks

Recurrent neural network (RNN) architectures such as Long Short-Term Memory (LSTM) are specifically good at simulating data that is sequential, incorporating data from time series, textual information, and even more. Due to the vanishing gradient issue, which involves typical RNNs have trouble detecting dependencies that are long-term in data. LSTMs were created to overcome some of these issues. By adding a memory cell that is capable of storing and retrieving data over a greater duration, LSTMs get around this restriction, making them appropriate for jobs like series prediction, detection of speech, and processing of natural languages.

The effective use of a cell state and a number of gates that control the transfer of information in and out of the cell state is the fundamental concept underpinning LSTM. There are three main gates in the LSTM architecture:

- *Forget Gate*: The Forget Gate determines whether to keep or discard the data from the prior cell state. It provides a forget gate output (f^t) between 0 and 1 for each element in the cell state from the current input (X^t) and the prior cell state (h^{t-1}) as inputs. Sigmoid activation function is used in its computation.
- *Input Gate:* What fresh information should be included in the cell state is decided by the input gate. Similar to the forget gate, it computes an input gate output (i^t) using a sigmoid activation function using the current input (X^t) and the prior cell state (h^{t-1}) as inputs. Additionally, it calculates a potential cell state (h^{t-1}) using the activation function of the hyperbolic tangent (tan^h).
- *Output Gate*: In order to create the current output and the following cell state, the output gate determines what data from the cell state should be used. It computes the output gate output (o^t) using a sigmoid activation function, taking the prior cell state (h^{t-1})), the current input (X^t), and the updated cell state (C^t). The tanh activation function is also used to generate the current output (h^t).

$$
\begin{aligned}
\mathbf{i}^{(t)} &= \sigma(\mathbf{W}_i\mathbf{x}^{(t)} + \mathbf{U}_i\mathbf{h}^{(t-1)}), && (1)\\
\mathbf{f}^{(t)} &= \sigma(\mathbf{W}_f\mathbf{x}^{(t)} + \mathbf{U}_f\mathbf{h}^{(t-1)}), && (2)\\
\mathbf{o}^{(t)} &= \sigma(\mathbf{W}_o\mathbf{x}^{(t)} + \mathbf{U}_o\mathbf{h}^{(t-1)}), && (3)\\
\tilde{\mathbf{c}}^{(t)} &= \tanh(\mathbf{W}_c\mathbf{x}^{(t)} + \mathbf{U}_c\mathbf{h}^{(t-1)}), && (4)\\
\mathbf{c}^{(t)} &= \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \tilde{\mathbf{c}}^{(t)}, && (5)\\
\mathbf{h}^{(t)} &= \mathbf{o}^{(t)} \odot \tanh(\mathbf{c}^{(t)}), && (6)
\end{aligned}
$$

The below is the illustrated figure of LSTM.

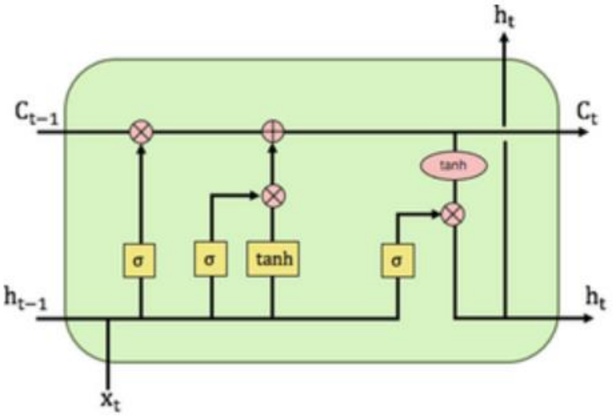


Figure 7:Long Short-Term Memory Unit

e) Artificial Neural Network

An Artificial Neural Network (ANN) is a type of computational model that is based on how the neural networks in the human brain are arranged and function. Pattern recognition, classification, regression, and other tasks can all be processed and learned from data using ANN models, which are made up of interconnected artificial neurons arranged into layers. Here is a quick explanation of how an ANN functions:

A. Layers and Neurons: Artificial neurons, also known as perceptron, are layered and make up ANNs. Typically, there are three sorts of layers: The first data or characteristics are received by the input layer. Hidden Layers: Through weighted connections and activation functions, these intermediary layers process and change the input data. ANNs are able to learn complex representations thanks to multiple hidden layers. The network's output, which might take the form of predictions, classifications, or regression values, is produced by the last layer.

B. Weights and Relationships: Each connection between neurons has a weight attached to it that indicates how strong the connection is. The performance of the network is optimised by adjusting these weights during training. Each neuron has its weighted sum of inputs (or weighted sum) calculated.

C. Activation Functions: Activation functions add nonlinearity to the model, enabling ANNs to recognise intricate patterns and connections in data. Sigmoid, ReLU (Rectified Linear Unit), and SoftMax (for multi-class classification) are often used activation functions. The weighted total of the inputs is added to the activation function to generate the output of the neuron.

ANNs train themselves using data through this approach. The weights of the network are iteratively changed during training to reduce the discrepancy between expected outputs and actual target values. Usually, optimisation procedures like gradient descent and backpropagation are used to do this.

f) AdaBoost

Machine learning applications that require classification and regression are typically handled by the ensemble learning technique known as AdaBoost, or Adaptive Boosting. It is a well-known and effective technique that integrates predictions from various weak learners (usually decision trees or stumps) to produce a robust and precise model.

A. Weak Learners: AdaBoost begins by learning from a base or weak learner, which is often a straightforward model like a decision stump (a decision tree with only one level). On the training data, this poor learner performs marginally better than random chance, but it is not very accurate.

B. Weighted Training Data: Each data point in the training sample is first given an equal weight. AdaBoost increases the weights of the data points that the weak learners from earlier iterations misclassified in subsequent rounds. AdaBoost can concentrate on the examples that are difficult to accurately categorise by emphasising the data points that were incorrectly categorised.

C. Sequential Learning: AdaBoost uses a method of sequential learning. It trains a new weak learner using the training data in each iteration. The correctness of the weak learner's performance on the training data influences how much weight it will have in the final model.

D. Weighted Voting: AdaBoost uses a weighted majority vote to combine the predictions of the weak learners. When making predictions, weaker learners with higher accuracy are given more weight. These weighted weak learners are assembled to create the final model.

E. Final Model: The predictions made by the weak learners are added up to create the final model. The model gives predictions from strong weak learner's larger weights and predictions from weaker learners lower weights.

2.6 Evaluation matrix

Metrics for evaluation are essential for assessing classification performance. The most widely used indicator for performance evaluation is accuracy. The percentage of those datasets that a classifier properly categorised on a given test dataset is the classifier's accuracy. Additionally, we use some other metrics to assess classification performance because the accuracy measure is insufficient for the text mining analysis to make a correct judgement. Three essential metrics that are frequently utilised are the F-measure, recall, and precision. We need familiarise ourselves with a few terms before diving into other measures:

- True Positive (TP)- indicates how many of the data points were successfully classified.
- True Negative (TN)- Indicate the number of data which are incorrect classified as negative.
- False Positive (FP)- represent the number of data which are correct but classified incorrectly.
- False Negative (FN) – represent the number of the incorrect data which are classified as positive.

a) Accuracy

The classifier's accuracy measures how frequently it appears to make the right prediction. It is calculated by dividing the number of accurate prediction by the total number of predictions.

$$Accuracy=\frac{No.of\ correct\ Prediction}{Total\ no.of\ prediction}$$

b) Precision

A classifier's precision can be used to evaluate how precise it is. False positives will be more prevalent with low precision than they will with high precision which will produce fewer of these errors. The definition of "precision" (P) is:

$$\text{Precision (P)}= \frac{TP}{TP+FP}$$

c)Recall

The sensitivity of a classifier is determined by recall. Fewer false negatives are caused by greater recall. The recall ratio is calculated by dividing the total number of correctly predicted occurrences by the total number of correctly identified occurrences. As an example of this

$$\text{Recall (R) -} \frac{TP}{TP+FN}$$

d)F measure

Precision and recall are combined to generate a single statistic known as F-measure (the weighted harmonic mean of recall and precision). This is how it is explained:

$$\text{F Measure} = \frac{2PR}{P+R}$$

3.Proposed Methodology

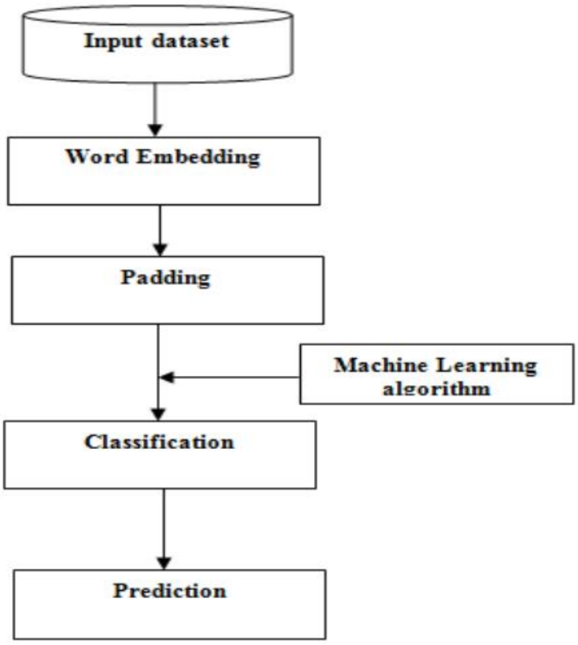


Figure 8: Proposed Architecture

We start with input dataset, which is used then for training and testing by splitting into two. We employ a word embedding model to discover the contextual relationships between words in the training set. Additionally, it improves how brief texts with limited content are represented. For this, we've used the embedding layer, one of the layers in the Keras module. This layer is used to perform the word embedding, which lowers the high dimensionality of the text. After this, padding will be performed ensure that each review is the same size so that processing the data is simple. Followed by this LSTM and different machine learning algorithms will be applied to the data for classification and predict the categories of unseen data then as a final step we evaluate the model to identify the one with better performance.

3.1 Data Collection

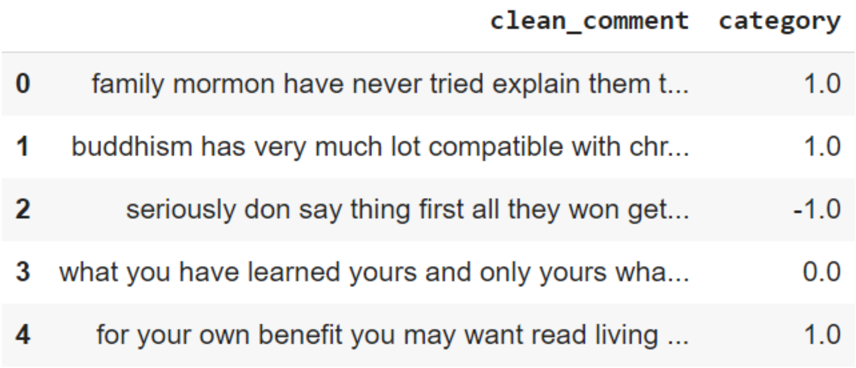
The dataset used in this dissertation is Twitter data set and reddit data set which are available in Kaggle [18]. The twitter data set consist of 162980 tweets whereas the reddit data set consist of around 37000 sentences for sentimental analysis. These Tweets and Comments were all obtained through the appropriate versions of PRAW and Apis Tweepy. All Tweets and Comments from Twitter and Reddit are cleaned using Pythons re and also NLP, with a Sentimental Label to each that goes from -1 to 1.

0- Indicating a neutral comment /tweet

1- Indication of a positive sentimental emotion

-1-Indication of a negative comment

For the purpose of sentimental analysis, the data set Twitter and Reddit is combined into a single data set. Then the entire data set is divided into testing and training for the model evaluation and fit. We use 70 % data for training and 30 % for testing.



	clean_comment	category
0	family mormon have never tried explain them t...	1.0
1	buddhism has very much lot compatible with chr...	1.0
2	seriously don say thing first all they won get...	-1.0
3	what you have learned yours and only yours wha...	0.0
4	for your own benefit you may want read living ...	1.0

Figure 9: Combined data set of Twitter and Reddit

3.2 Exploratory data Analysis

Exploratory data analysis is a crucial technique for doing first studies on data in order to identify trends, identify discrepancies, evaluate hypotheses, and triple-check presumptions with the help of data summary statistics and visual representations.

This section tries to investigate the data's contents and comprehend its behaviour in relation to its labels.

Started by combining both dataset into a single data set. To provide variation in the training set and enhance the overall performance of the model, both sets of data were combined rather than one being used for training and the other for testing. Our model will be more diverse when the data is more diverse.

3.2.1Visualization

a.Bar Plot

This gives us the total count of neutral, positive and negative comments present in the dataset. This plot provides the insight that, the presence of positive comments is more than neutral and negative.

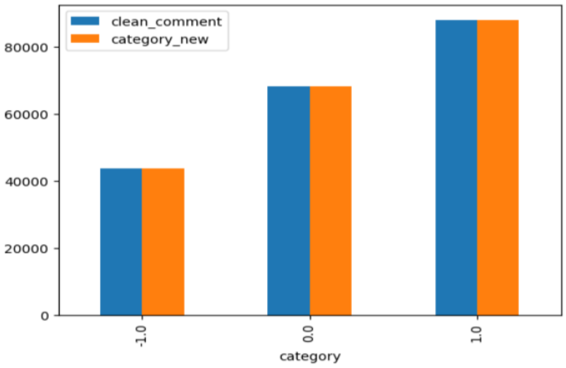
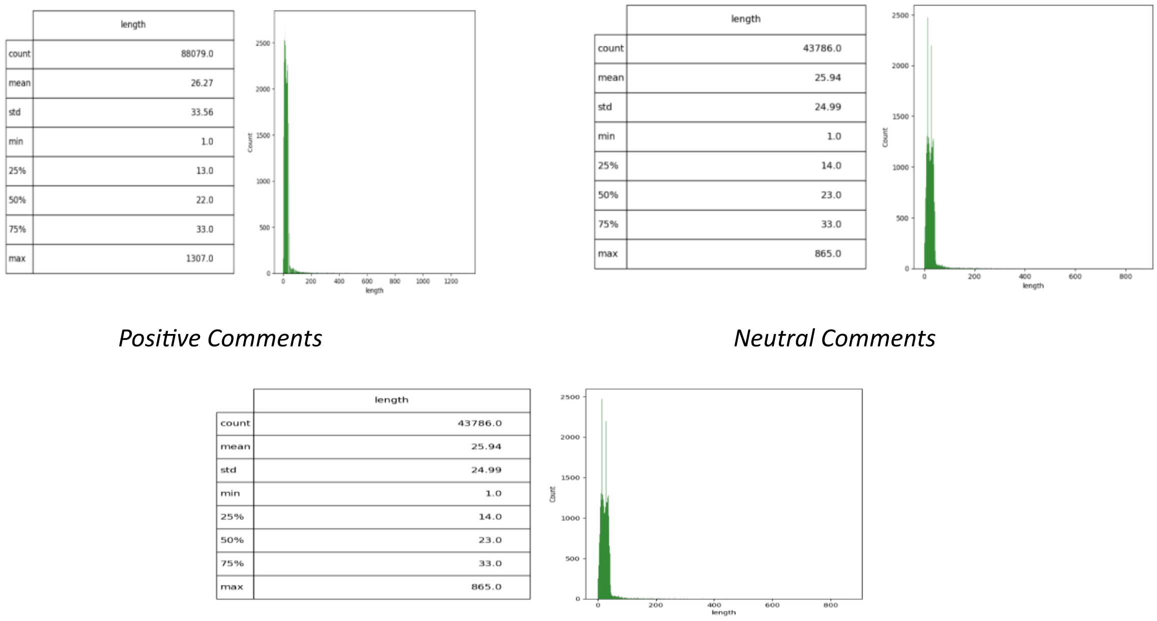


Figure 10: Bar plot of categories

b. Hist plot

In this plot, we plotted a table and histogram plot for each category, table give us the information like mean, median, standard deviation and quartile range of the particular category of emotions. From the distribution graph it is clear that positive comments have higher length, which follows the negative and then neutral.



	length
count	88079.0
mean	26.27
std	33.56
min	1.0
25%	13.0
50%	22.0
75%	33.0
max	1307.0

Positive Comments

	length
count	43786.0
mean	25.94
std	24.99
min	1.0
25%	14.0
50%	23.0
75%	33.0
max	865.0

Neutral Comments

	length
count	43786.0
mean	25.94
std	24.99
min	1.0
25%	14.0
50%	23.0
75%	33.0
max	865.0

Figure 11: Distribution graph and table

C. Top Word Distribution

This plot creates a bar plot with the frequency of top 30 words and plot them in the order of word having highest frequency to lowest frequency. The y- axis in the plot represents the word count whereas x -axis denotes the top words in the dataset.

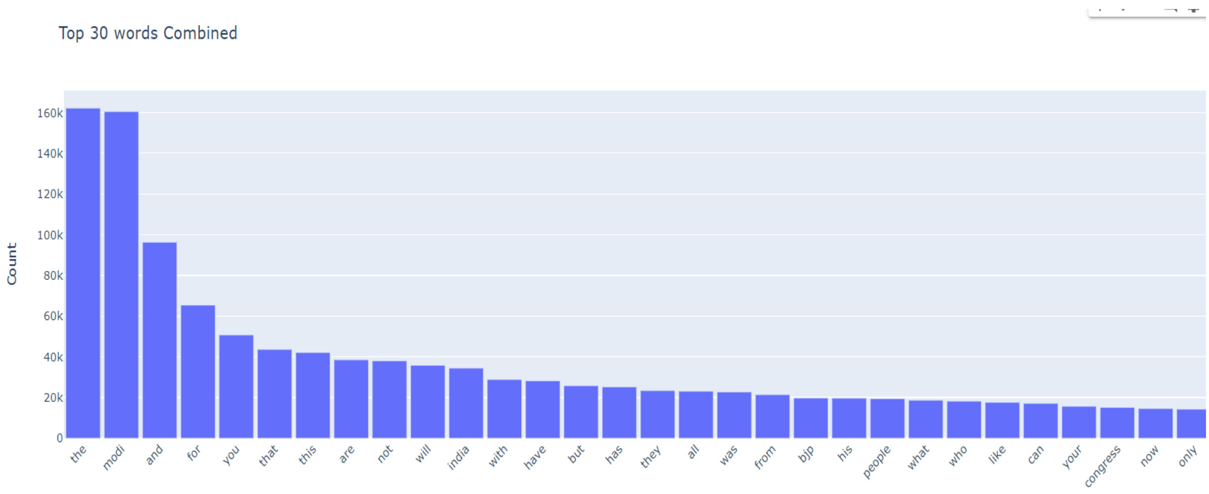


Figure 12: Top Word Count Distribution

d. Target Distribution Graph

This plot is drawn to analyse the data to see how values of a categorical variable are distributed, use a distribution plot (using Seaborn). By looking the graph around 60k falls in the category of negative and more than 10k is in neutral and positive category have the highest count distribution.

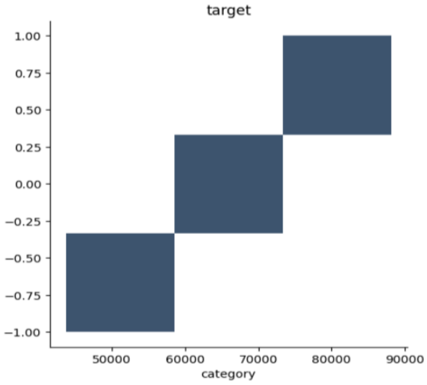


Figure 13: Target distribution graph

e. Pair Plot

Each variable in the dataset is plotted against every other variable in a grid of scatterplots created by a pair plot. The grid will contain 'n' rows and 'n' columns if you have 'n' numerical variables. Typically, the histograms or kernel density estimate for all variables are displayed on the diagonal of the panel. Pair plots are excellent for exploring the relationships between variables visually. The plot reveals that feature present in the data set like length, category and word count is having a weak positive linear relationship between the variables.

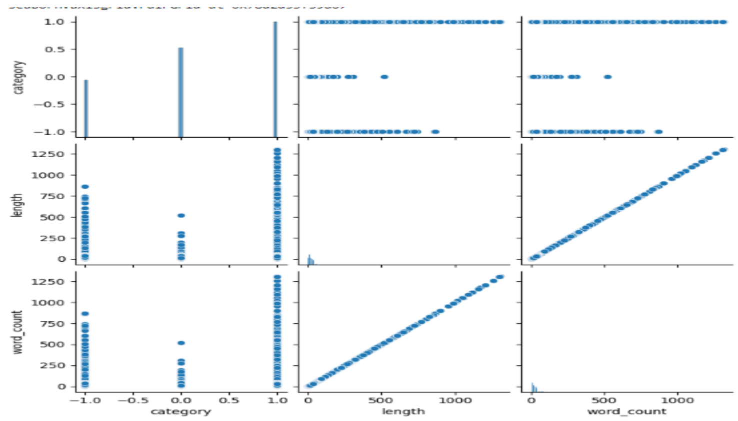


Figure 14: Pair plot of features

f. Heatmap

Heatmaps are frequently used to display a dataset's correlation matrix. The correlation coefficient between two variables is represented by each cell in the heatmap in this instance. High positive correlations are commonly illustrated using one colour, but high negative correlations are typically illustrated using a different colour. This aids in determining whether variables are inversely or strongly connected.

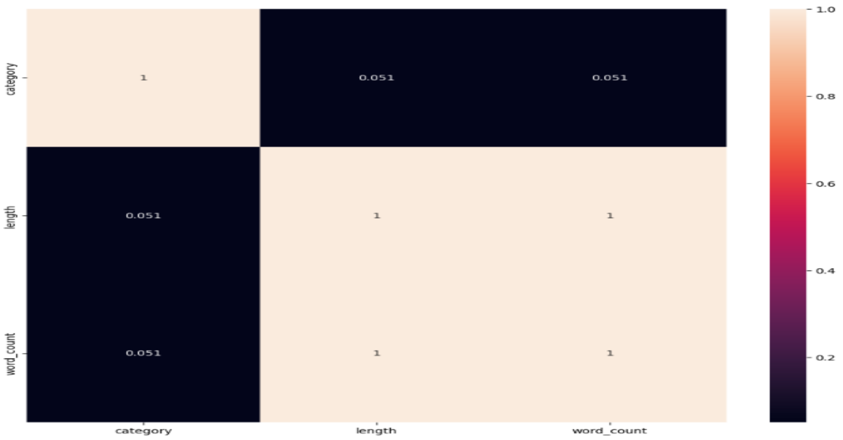


Figure 15: Heatmap of data

3.3 Data Preprocessing

Data preprocessing refers to the procedures we must follow to change or encrypt data so that a machine can quickly and readily decode it. The algorithm must be able to quickly assess the properties of the data in order for the model to produce precise and accurate predictions. Individuals also prefer to utilise a lot of acronyms, slang, emoticons, and other non-wordy expressions to try to convey as much emotions as possible. Additionally, we can utilise the '@' sign to refer to people and the '#' symbol to assign tags on Twitter. The sentiment classification does not really benefit much from this information. Therefore, the initial preprocess for each sentence is as follows:

a) Pandas Library: This tool was used to make it easier to manage input files containing publications' parts. Then, a data frame was generated just with the elements that were crucial for the research, meaning that records with inaccurate values or unrated records were deleted.

b) Regular Expressions: These were used to get rid of the URLs and username references from other users. Alphanumeric characters that fit a predetermined search pattern might also be located, replaced, or removed, as well as extra spaces. Additionally, numeric characters that did not help with sentiment analysis and the duplication of suffix characters that were utilised for emphasis reasons were eliminated.

c) Emoticons: Emoticons are symbols, similar to semicolon and brackets of some kind that produce renderings of human face gestures, such as a joyful person:-), as well as other images that are essential to comprehending the emotions associated with every journal. In order to express feelings and impressions quickly, emoticons frequently appear on online platforms, particularly on Twitter. As a result, a collection of regular expressions that mainly contained these representations was created. Users' invented words can be replaced by a list of frequently used unofficial acronyms was also created. The LOL term, for instance, was replaced with its corresponding complete form, laugh out loud.

d) Autocorrect Library: When a word is entered, the autocorrect library compares it to a list of words from reputable dictionaries to determine how similar it is. The entire input word is returned if it has been spelt correctly. If it is misspelt, the words in the list are evaluated to see how similar they are, and if they are more similar than a predetermined threshold, the word in the list is substituted. If not, it is returned unaltered and in its entirety.

e) Emoji Library: These Unicode characters are icons that depict a variety of things and face expressions. This feature sets our work apart from competing research because most studies do not account for emoticons and emojis. However, opinion mining employs them in the majority of publications. The Unicode Consortium's mapping list is used by this library. If any of the Unicode characters found in the content are found to be on that list, they are substituted with the text format of its representation.

f) Part-of-Speech Tagging: Each word in the text is labelled with the part of speech (adverb, verb, object) that it belongs to. This method evaluates and assigns the part of speech to the specific term being investigated by taking into account both the context of the text to be analysed and a collection of aggregated parts (corpus).

g) Lemmatization: Lemmatization is the process of removing complex suffixes and retrieving the term's lexical form by taking into account lexical and morphological analyses of words. It is used following POS tagging and uses machine learning algorithms to make emotional analysis easier. The POS tagging labels used in this work are in Penn Treebank format.

h) Tokenization: Using this technique, sentences are split across a group of symbol words which can then be combined to recreate the original content. Text characters created from emoticons and other emoticons can be used as symbols instead of being broken up into individual letters or punctuation. All phrases are subjected to the tokenization procedure, and the terms are all kept in the same token list. In essence, each post generates a list of tokens. More conversions are made when the publication's details are now listed in the order that they appear, which optimises and enhances the viewing process.

i) Hashtag Removal: A hashtag is a term that begins with a '#'. Hashtag distinguishes itself from other words by adding a topic or tag to the tweet and comment. Typically, the tag refers to the subject matter mentioned in this tweet rather than the opinions of the people involved. This term may include some information, but it's not very significant. Therefore, I choose to leave the term in tact but only delete the '#' sign, treating the tag like any other word in a sentence.

j) Punctuation: These are additionally list tokens. Punctuation marks are typically eliminated since they don't add any emotional weight to the writing.

k) Vectorization: Natural language processing (NLP) and information retrieval use the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique to transform text input into numerical vectors. In this procedure, a corpus of documents is converted into a matrix of numerical values, where each row corresponds to a document and each column to a distinct term (word or token) from the corpus as a whole. For text-based machine learning applications, TF-IDF vectorization is especially helpful

l) Stop words: These are words that frequently emerge without expressing any sort of emotion. Such terms have been eliminated because the goal of the entire project is to study meaningful phrases in order to ascertain the overall mood conveyed in publications. Why are stop words being eliminated? It's because these stop words don't mean anything when used in conjunction with sentences, thus they aren't necessary when we construct the sentences' meanings.

Figure 16: Word Cloud

3.4 Algorithm Implementation

1) Long Short-Term Memory

At this stage, the LSTM algorithm is used to create a model for categorising the reviews according to sentiment. With the help of the training dataset, the model is trained before being put to the test.

Our model consists of 4 layers starting with an Embedded layer, then spatial dropout, an LSTM layer, and a softmax output layer. Processing data sequences, such text or time series, frequently involves the usage of an embedding layer. It transforms input sequences using integer encoding into dense vectors with predetermined sizes. The built-in model converts input integers from a 1000-word vocabulary to 100-dimensional dense embeddings. For convolutional and 1D sequence models, there is a version of dropout called spatial dropout. In order to avoid overfitting, a portion of the input units are randomly set to zero during training. The dropout rate is set at 0.2, which means that during each training phase, 20% of the input units are zeroed out. The third layer is LSTM which is a Recurrent neural network (RNN) that good at handling sequential data. It contains 100 units and introduces regularisation to the LSTM layer by setting dropout and recurrent dropout to 0.2. The last layer is a dense layer with three output units that is fully connected. SoftMax, a popular activation function for multi-class classification tasks, is applied in this case. The final layer outputs from the model are transformed into probability distributions across three classes.

The visual representation of the modelled neural network is depicted in the figure 17 given below. This shows the overall architecture of the model created, showing the various layers, and shapes of the tensors flowing between them.

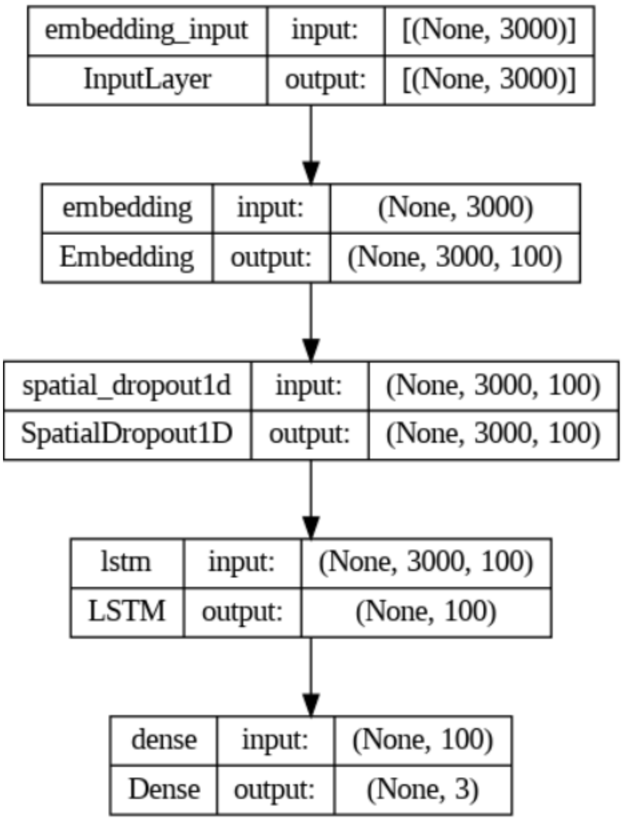

embedding_input | input: | [(None, 3000)]
InputLayer | output: | [(None, 3000)]
embedding | input: | (None, 3000)
Embedding | output: | (None, 3000, 100)
spatial_dropout1d | input: | (None, 3000, 100)
SpatialDropout1D | output: | (None, 3000, 100)
lstm | input: | (None, 3000, 100)
LSTM | output: | (None, 100)
dense | input: | (None, 100)
Dense | output: | (None, 3)

Figure 17: Visual Representation of Model

When training a learner using an iterative technique like gradient descent, early stopping, a sort of regularisation, helps prevent overfitting. These methods update the learner after each repeat to enhance its fit with the training data. Followed by this the model is fitted to the training data with a total of 5 epochs, a full traverse across the entire training dataset constitutes one epoch. The validation split ratio is 15% the model's performance on this validation set will be monitored. We set patience as 3 which indicate if the validation loss does not decrease for three successive epochs, training will end. After fitting the model, the model is evaluated using various performance measures.

2)Naïve Bayes

The next model built for the sentimental analysis classification is done with the help of machine learning model called Naïve Bayes. After performing all the necessary preprocessing required for the data set, we fit the training data to the multinominal naïve bayes model. The model trains the TF-IDF vectors of the training vectors and then predict the emotional category. Then assesses the model's performance for each class using accuracy and F1 ratings. The F1-scores provide information on the model's performance for each class, particularly in a multi-class classification scenario, whereas the accuracy provides an overall assessment of how well the model is performing.

3)Decision Tree

To evaluate which model performs better in the task of classifying the tweets and comments, another model called decision tree algorithm is also implemented. Like naïve bayes algorithm, we implement the same logic for decision tree like train the model with TF-IDF vectors. To improve the further performance of the model, hyper tuning is done. This method works by fine tune the different parameters of the decision tree or by using other tree-based algorithms like random forest etc. Here the hyper tuning is done by applying maximum depth, minimum sample split, minimum sample leaf and entropy with different values and one with better performance will be selected and used to fit the model.

4)Support vector Machine

Another model which is used for the performance comparison is support vector machine. Linear support vector machine is used which is used for multiclass and binary classification problem. It is trained on a labelled training dataset to discover the appropriate decision boundary for classifying the data into distinct groups. The model identifies a linear hyperplane during training that increases the distance between classes. It is used for predicting the labels of a test data set after training. By computing and reporting the model's accuracy, which measures the percentage of correctly categorised samples in the test data, the code assesses the model's performance. The F1 score, a metric that balances precision and recall for each class in a multi-class classification task, is also computed and printed.

5) Artificial Neural Network

This simple artificial neural network consists of 3 layers, having an input layer, two hidden layers with ReLU activation functions, and an output layer with SoftMax activation for multi-class classification. The Adam optimizer, sparse categorical cross-entropy loss (suited for integer-encoded labels), and accuracy are used during the model compilation process. To track validation performance during training, the model is trained using training data for 10 epochs, a batch size of 64, and a validation split of 20%. Finally, a different test dataset is used to evaluate the trained model, and the test accuracy is shown.

6) AdaBoost

Adaptive boosting is known as AdaBoost. It uses an ensemble method to integrate the shallow decision trees' predictions with those of other weak learners to produce a robust and precise model. Then trained the model using training data. All training examples are given equal weights by AdaBoost during training, and the base estimator (decision tree) is fitted to reduce mistakes. The significance (weight) of examples that were incorrectly identified is then increased in succeeding rounds.

The model uses a Decision Tree Classifier with a maximum depth of 1 as the base estimator. This decision tree is weak at learning since it only relies on one feature at each level to make basic binary judgements, which makes it relatively shallow. After training, the AdaBoost Classifier makes predictions on new information using the ensemble of decision trees, then integrates these individual predictions to get a conclusion.

4. DISCUSSIONS OF RESULT

This section includes a discussion of the outcomes as well as a thorough description of the evaluation metrics.

4.1 Experimental Results

This study is conducted using 3 class problem namely positive, negative and neutral. The data is collected from 2 different social media platform -Twitter and Reddit. Each of these reviews contains a sentiment category rating, in which 0 indicate neutral, 1 for positive and -1 for negative comments. The dataset is divided into 70:30 ratio for the purpose of training the model and for testing. In this study, various machine learning algorithm like Naïve Bayes, Support vector machine, Decision tree classifier, adaboost and some deep learning technique like Long Short-Term memory networks, Artificial neural network was implemented. Several other preprocessing methods like Tokenization, Segmentation, Normalization are carried out on the data set. Both the train set and the test set are converted into distinct term document matrices (TDM), after which each word's frequency is recorded. In order to represent the data, the TF-IDF is used. The TDM is then utilised as an input for a variety of machine learning classifiers after that.

The figure 18 represents the collection of negative, neutral and positive sentiments obtained from the Twitter and Reddit data.

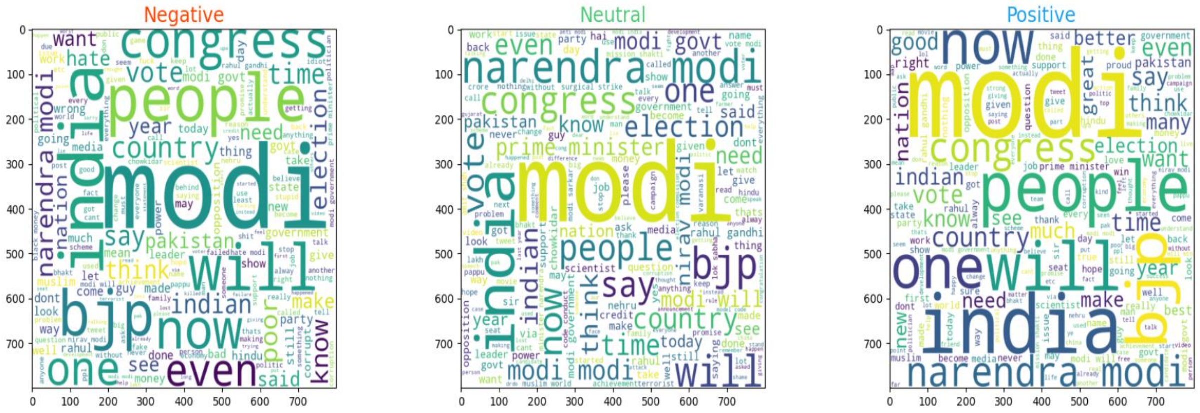


Figure 18: Collection of different sentiments

The LSTM model has an accuracy of 83% for classifying the emotions. Figure 19 depicts the model's accuracy and loss for each iteration over five epochs.

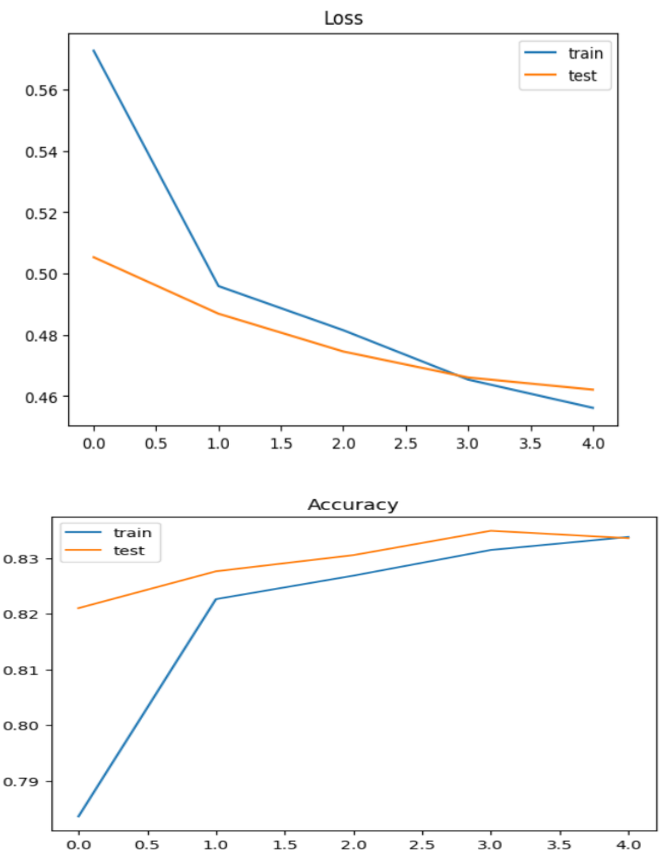


Figure 19: LSTM model performance- Loss, Accuracy

The confusion matrix for the classifiers that are implemented is given below in Figures.

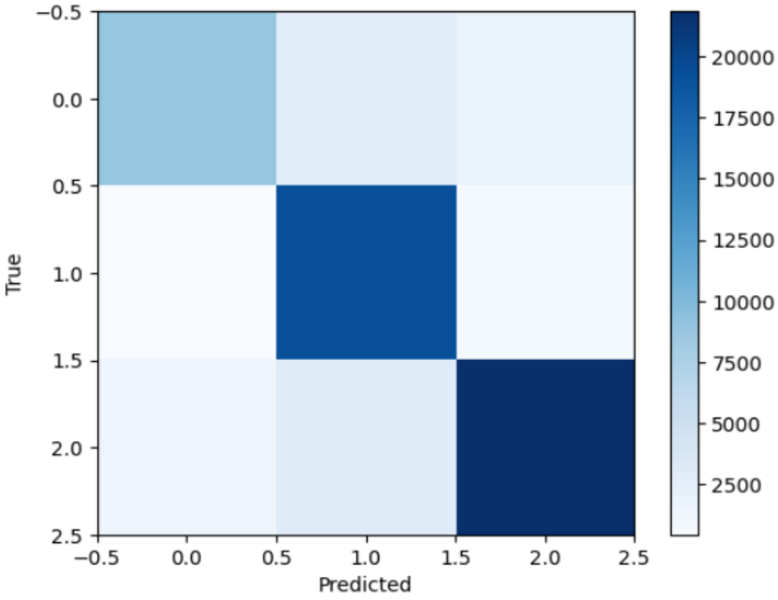


Figure 20: Confusion matrix of LSTM

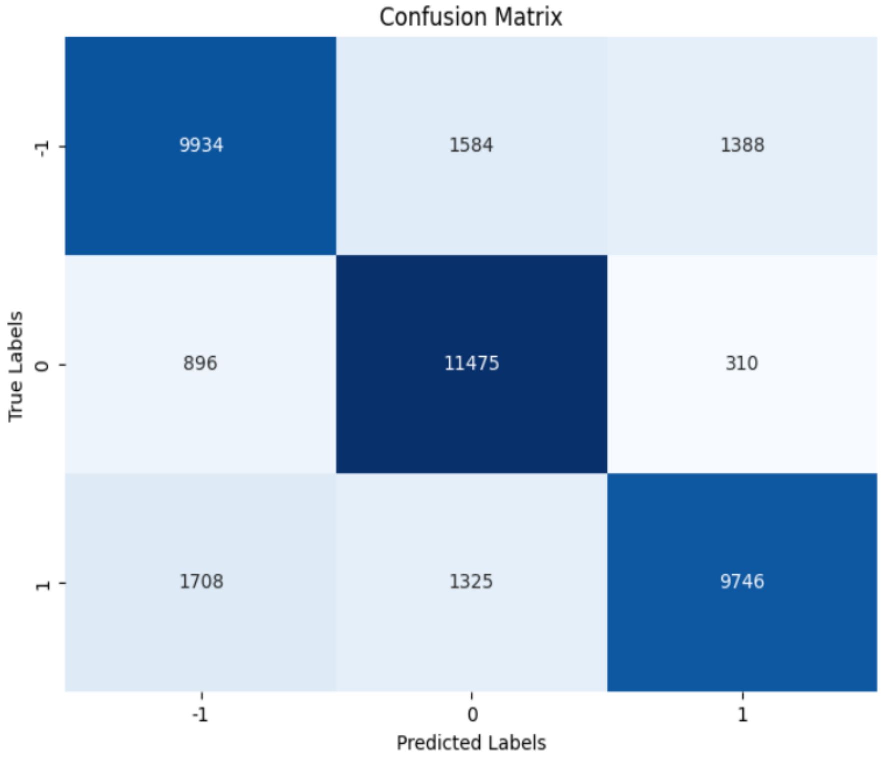


Figure 21:Confusion matrix of SVM

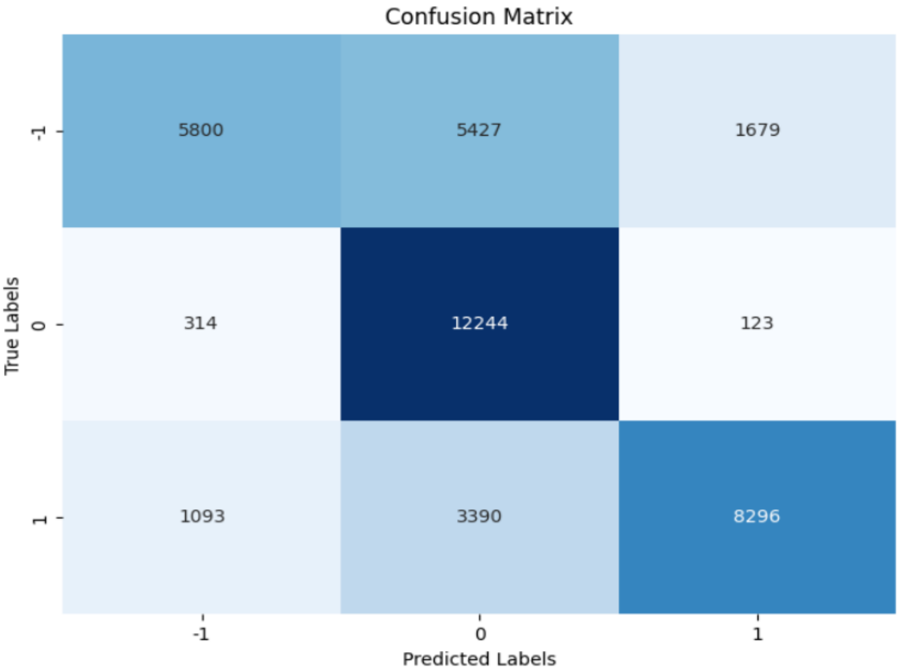


Figure 22:Confusion matrix of Adaboost

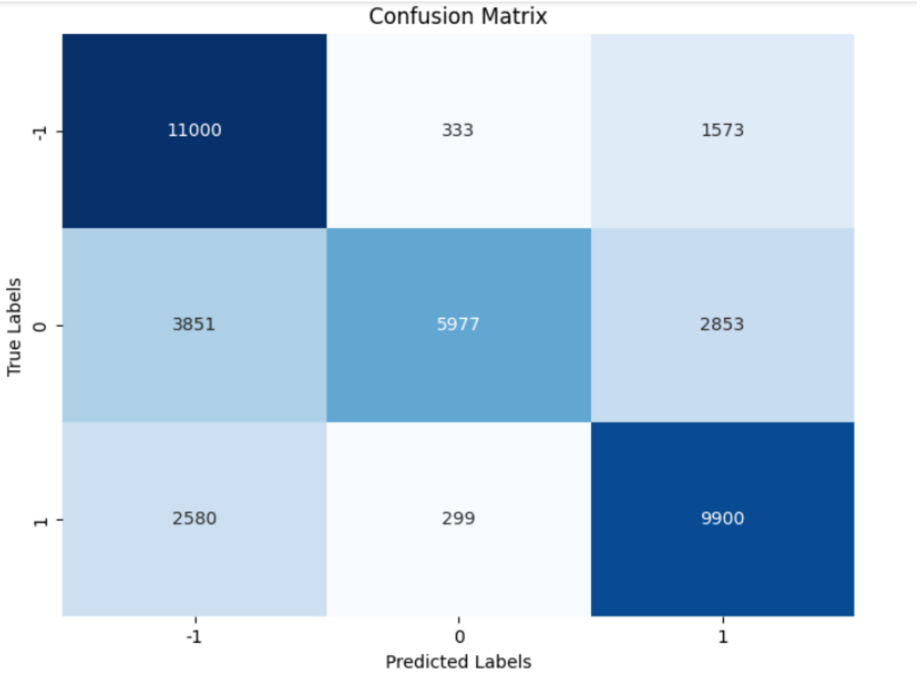


Figure 23:Confusion matrix of Naïve Bayes

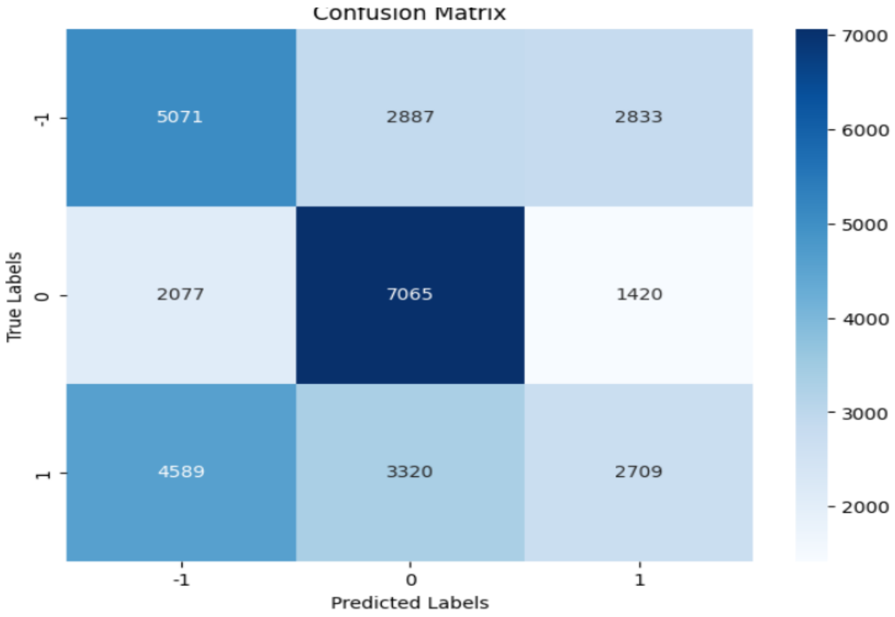


Figure 24 : Confusion Matrix of Artificial Neural Network

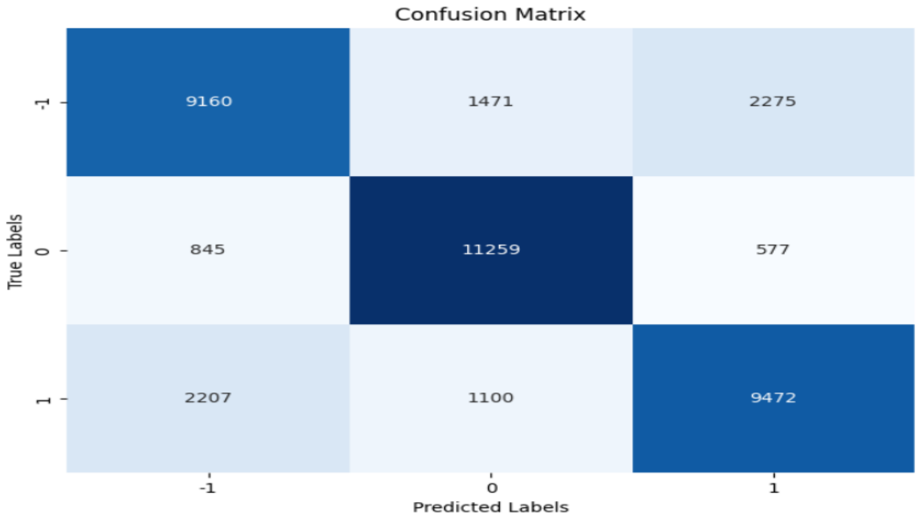


Figure 25: Confusion Matrix of Decision Tree

We can see the incorrectly categorised positive, negative, and correctly classified positive, negative reviews in the confusion matrix. The negative and positive numbers in the principal diagonal are accurately identified as such.

The experimental result of all the implemented classifiers is shown in the below table Table 1. We can see that LSTM outperforms than the other methods in the three- classification problem. The highest accuracy obtained from all these models was 83%. LSTM performs exceptionally well in classifying sequence text data.

Table 1: Experimental result of classifiers

Classifier	**Accuracy**	**Precision**	**Recall**	**F1- Score**
LSTM	0.83	0.83	0.81	0.81
Support Vector Machine	0.81	0.81	0.81	0.81
Naïve Bayes	0.70	0.74	0.70	0.69
Decision Tree	0.78	0.78	0.78	0.78
AdaBoost	0.69	0.74	0.69	0.68
ANN	0.46	0.45	0.46	0.45

The next good model among the classifiers is support vector machine with an accuracy of 81% and least performance is shown by Artificial Neural network.

5. Project Management and Reflection

5.1. Project plan Discussion

The research problem of analysing the sentimental emotion behind the Twitter and Reddit data will be using LSTM.

Starting by data cleaning and processing dataset to ensure the data will be in a suitable format for the further analysis. Following this, next step is the development the LSTM network, including the number of layers, hidden units, and input/output configurations to predict the correct emotions. Later in this stage will experiment with different configurations and hyperparameters to optimize model performance.

Once this is done, then evaluate the model's ability to generalize across different domains by testing it on separate datasets related to various topics. Measure the performance on these datasets and analyse the model's adaptability and transfer learning capabilities.

As the last stage of the research, planning to conduct an error analysis to identify common misclassifications or challenges faced by these models. This analysis can provide insights into areas for improvement and potential modifications to the model architecture or training process.

Work package	**Completion Date**	**Length of days**
WP1- Data collection and set up	July 5	1 day
WP2- Literature Review	July 16	10 days
WP3- Exploratory Data analysis	July 21	5 days
WP4- Preprocessing	July 27	6 days
WP5 – Model selection	August 5	7 days
WP6-Training and Validation	August 15	9 days

WP7- Evaluation and Error Analysis	September 1	15 days
WP8 – Result Analysis and Dissertation	September 15	10 days

5.2. Work Progress Assessment

I began working on the project according to the established plan, collecting all the necessary data on July 5th. All tasks proceeded as scheduled, and activities such as preprocessing and exploratory data analysis were completed two days ahead of the anticipated date. However, the project came to a halt during WP6 for a period of two weeks due to my need to retake one subject, which subsequently extended the submission deadline. Following my successful completion of the exams, I resumed the project on September 1st, making adjustments to the plan as needed.

5.3 Reflections

In my journey of gaining practical experience in deep learning and machine learning for sentiment analysis. At the outset, my lack of abstraction resulted in convoluted code. I was prone to simply writing code as ideas emerged, without a comprehensive architectural plan. Looking back, I recognize the importance of establishing an overarching architecture from the project's inception, even if it may not have seemed entirely coherent at that early stage. This architectural foresight has proven valuable for the ongoing refinement and enhancement of the sentiment analysis model.

6. Conclusion and Future Work

Sentiment analysis at the level of reviews has successfully eliminated the remarked common alternatives and evaluation terms from the input dataset. LSTM, which does a great job of categorizing the reviews, is used to develop the model for interpreting the emotion of the comments and tweets from the platform of Twitter and Reddit. By learning which content is well received by the audience, this tactic will help to understand the sentimental emotions behind that. The study examines and refers to a number of sentiment mining and classification systems. Our findings demonstrated that, in terms of accuracy, the Long Short Term Memory Networks algorithmic standard outperformed others. In this study, a sentiment classification model is proposed that may be used to categories the Tweets and comments from Reddit according to their emotional content in sequence of text data. This system is unique and more effective than others since it can identify long sequence data with the aid of LSTM, which uses long-term memory and is therefore particularly effective at handling long-term dependencies. Even with larger datasets, the model's performance can be enhanced further.

The research scope can be broadened by including other feature selection strategies, such as mutual information (MI), information gain, and chi-squared test, for better representation. We can mix hybrid classifiers like SVM with additional techniques to increase accuracy. A good suggestion approach can be created by taking into account the feelings that customers' comments evoke. Although the suggested LSTM-based strategy is the best, we can use the REST API-based web service approach to evaluate real-time data and classify them into various moods. An analysis of the enormous volume of tweets on a big data platform can address scalability difficulties.

REFERENCES

[1] Temple, Krystal. "What Happens in an Internet Minute?" Inside Scoop (2012). A

[2] Bollen, Johan, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market." Journal of Computational Science 2.1 (2011): 1-8.

[3] Yang Gao, Christian M. Meyer, Iryna Gurevych."APRIL: Interactively Learning to Summarise by Combining Active Preference Learning and Reinforcement Learning."

[4] Elias Dritsas, Gerasimos Vonitsanos, Ioannis Livieris. “Pre-processing Framework for Twitter Sentiment Classification”(2019)

[5] Yoon Kim.” Convolutional Neural Networks for Sentence Classification”.

[6] Jeffrey Pennington, Richard Socher, Christopher D. Manning.” GloVe: Global Vectors for Word Representation”.

[7] Jiacheng Xu, Danlu Chen, Xipeng Qiu, Xuanjing Huang”Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification”.

[8] Zichao Yang, Diyi Yang, Chris Dyer , Xiaodong He, Alex Smola , Eduard Hovy.” Hierarchical Attention Networks for Document Classification”

[9] Robin Jia, Percy Liang.” Adversarial Examples for Evaluating Reading Comprehension Systems”.

[10] Andreas Kanavos, Nikoalos Nodarikis, Giannis Tzimas Large Scale Implementations for Twitter Sentiment Classification”. (2017)

[11] Taboada, Maite, et al. "Lexicon-based methods for sentiment analysis." Computational linguistics 37.2 (2011): 267-307.

[12] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." Proceedings of the ACL-02 conference on Empirical methods in natural language processing Volume 10. Association for Computational Linguistics, 2002.

[13] McCallum, Andrew, and Kamal Nigam. "A comparison of event models for naive bayes text classification." AAAI-98 workshop on learning for text categorization. Vol. 752. 1998

[14] Yousef, Ahmed Hassan, Walaa Medhat, and Hoda Korashy Mohamed. "Sentiment Analysis Algorithms and Applications: A Survey." (2014).

[15] Mullen, Tony, and Nigel Collier. "Sentiment Analysis using Support Vector Machines with Diverse Information Sources." EMNLP. Vol. 4. 2004.

[16] Dobra, Alin. "Decision Tree Classification." Encyclopaedia of Database Systems. Springer US, 2009. 765-769

[17] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., &Stede, M.. “Lexicon based methods for sentiment analysis”. Computational linguistics, 2011:37(2), 267-307.