DEEP LEARNING-BASED SENTIMENT ANALYSIS OF AMAZON KINDLE STORE REVIEWS USING A HYBRID 3-CONVOLUTIONAL CNN AND 2-GRU LAYERS RNN MODEL.

MAJOR PROJECT REPORT

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF DEGREE OF

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

KISHAN SONI

2K21/CSE/11

Under the supervision of

Dr. MANOJ SETHI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi – 110042

2021-2023

M.Tech (Computer Science & Engineering)

KISHAN SONI

2023

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Kishan Soni, 2K21/CSE/11 student of MTech (CSE), hereby declare that the project dissertation titled "**Deep learning-based sentiment analysis of Amazon Kindle store reviews using a hybrid 3-Convolutional CNN and 2-GRU layers RNN model.**" which is submitted by me to the Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi KISHAN SONI

Date: 25/05/2023 2K21/CSE/11

COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**Deep learning-based sentiment analysis of Amazon Kindle store reviews using a hybrid 3-Convolutional CNN and 2-GRU layers RNN model.**" which is submitted by Kishan Soni, 2K21/CSE/11 Computer science and engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this university or elsewhere.

Place: Delhi **Dr. MANOJ SETHI**

Date: 25/05/2023 **DTU CSE**

ACKNOWLEDGEMENT

I am extremely grateful to my project guide, Dr. Manoj Sethi, Professor, Department of Computer Science and Engineering, Delhi Technological University, Delhi for providing invaluable guidance and being a constant source of inspiration throughout my research. I will always be indebted to him for the extensive support and encouragement she provided. I am highly indebted to the panel faculties during all the progress evaluations for their guidance, constant supervision and for motivating me to complete my work. They helped me throughout by giving new ideas, providing necessary information and pushing me forward to complete the work.

KISHAN SONI
2K21/CSE/11

ABSTRACT

Sentiment Analysis is a part of Natural Language Processing (NLP) where we try to train a machine in such a way that it generates the ability to define the overall opinion about certain context such as negative, neutral, or positive. Data is taken and various pre-processing steps are applied and tagging of data is done to define its orientation then this minimalized data is converted into vector space as machine understand numbers not text using sentiment score or weightage to each word or frequency count methods. Then various machine learning algorithms are applied, and results are evaluated.

In this work we have taken a different data set and has classified the reviews as positive and negative easily which can help company to see negative reviewed products without reading too much reviews. Focus of this work is negative rated products. Data is first preprocessed by removing the null values and feature is extracted using vectorizer. Model is trained with balanced data, 80% of the data is used to train the model and 20% data is used to test. Further the test accuracy is obtained using different classifiers.

To capture the complex relationships among users and comments, we employ GNN to model the graph structure inherent in the dataset. By leveraging the graph representation, our approach achieves improved sentiment classification accuracy compared to traditional methods. The results demonstrate the effectiveness of GNN in capturing nuanced sentiment patterns within Weibo comments, offering valuable insights for understanding public sentiment on social media platforms.

For dealing with imbalanced data the Support Vector Machine (SVM) algorithm is combined with Particle Swarm Optimization (PSO) and other oversampling methods to create a hybrid strategy. The data was obtained with the assistance of Jeeran, a well-known Arabic assessment social network. To correct the feature weights, a PSO strategy is used, and four distinct oversampling methods—Synthetic Minority Oversampling Technique (SMOTE), SVM-SMOTE, Adaptive Synthetic Sampling (ADASYN), and borderline-SMOTE—are utilized to correct the dataset's imbalance and produce an optimized dataset. In terms of accuracy, F-measure, G-mean, and area under the curve (AUC), the proposed PSO-SVM method performs better than previous classification algorithms for a variety of dataset versions.

Further we tried deep learning models. In this particular CNN model, we utilize an embedding layer with a dimension of 64, followed by three convolutional layers. For the RNN model, we are opting for a simple architecture. It consists of an embedding layer, two layers of GRU, followed by two dense layers, and ultimately the output layer. To optimize performance, we employ the CuDNNGRU instead of the regular GRU, as it offers significantly faster computation, potentially over ten times faster. The maximum accuracy score achieved on the testing data set is 95.02% with the highest F1 score of 95.04% on negative review predictions. Highest accuracy is achieved by the hybrid model.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SVM	Spatiotemporal Data Mining
CNN	Convolutional Neural Network
GRN	Gated Recurrent Unit
RNN	Recurrent Neural Network
GCN	Gated Recurrent Unit
BERT	Bidirectional Encoder Representations from Transformers
LSTM	Long Short-Term Memory
PSO	Particle Swarm Optimization
Bi-LSTM	Bi-Directional Long Short-Term Memory
API	Application Programming Interface
ML	Machine Learning
DL	Deep Learning

CHAPTER 1

INTRODUCTION

1.1. GENERAL

Sentiment Analysis is the natural language processing for understanding an attitude, thought, or judgment of the public about a particular product or topic. Opinion mining is a part of sentiment analysis, where you collect, preprocess, and analyze the data about product such as review or opinion on product present on social media sites, shopping sites, blogs, or tweets. It can be used in several different ways.

The rapid digitization in the last decade has caused people to use the internet for completing all the basic tasks that they used to do manually. Since e-commerce websites display the products using images, it is hard to figure out the quality of the product's quality. As a result of this, there is a feature on internet that allows the user to write a review and rate a product. Rating of the product help us to understand the sentiment of people whether they are liking the product or not and review writing gives us the idea of positives and negatives of the product.

Due to increase in usage of e-commerce number of sellers selling their product has increased asking customer to review their product. Large number of reviews are available for a most used product. Because of this, it is difficult for an individual to read them and decide whether to purchase the product or not. To handle this sentiment analysis is used. Using sentiment analysis one can easily understand the emotion of the content [4].

Web users have a platform to share their thoughts and opinions on a wide range of topics and events through social media. For example, on performing sentiment analysis on

Twitter, companies can evaluate and analyze the market, or tweets posted by general mass about them, and competitors. Lot of online platforms are available on which one can perform sentiment analysis.

Nonpolar, positive, or negative sentiments are associated with textual content. Sentimental analysis is a type of text analysis that measures the sentiment of an article. Therefore, sentimental analysis has the primary objective of identifying the polarity of a textual content and, therefore, assessing its classification.

Sentiment analysis has many applications in various industries. As it helps in understanding the online conversation between two people or public opinion, companies use sentiment analysis in doing market research and figuring out if their customers like a particular product (or service) or not. Then, according to the findings of the sentiment analysis, the organization can modify the respective product or service and achieve better results.

There are various levels at which you can perform a sentimental analysis, including aspect, sentence, and document levels [3].

1. Document level

Sentiment of whole Document is evaluated into positive, negative, or neutral sentiment for a product or service.

2. Sentence level

Sentiment Analysis of sentence is done. Here sentiment classifier classifies statement as nonpolar, positive, or negative. It is basically used in review sentiment analysis or comment analysis.

3. Entity and Aspect level

Entity level or Aspect level sentiment analysis works on the aspect or feature of the product.

Here, we give sentiment or opinion on feature or aspect of the product which helps the customer to understand the liking or disliking of different feature of product like size, build quality, weight of camera. We can divide Aspect level sentiment Analysis as:

- **Aspects:** description of quality, quantity, size, color, and all other features of product.
- **Sentiments:** positive or negative opinions about a product feature.

1.2 PROBLEM FORMULATION

With the continuous growth of the e-commerce industry, online shopping has gained significant traction, surpassing traditional in-store experiences. As a result, customer feedback and experiences have become increasingly important. Potential buyers heavily rely on customer reviews, ratings, and feedback to assess the quality and value of products or services.

In contrast to physical stores, where customers can directly interact with items, online shoppers depend solely on website information and shared experiences. This shift in consumer behaviour underscores the significance of the customer's voice. Customer feedback and reviews play a critical role in shaping the perceptions of prospective buyers and influencing their purchasing decisions.

Consequently, it is vital for businesses to actively engage with and address customer needs and expectations. By attentively listening to customer feedback, companies can gain valuable insights into their strengths and weaknesses. This empowers them to make strategic decisions that align with customer preferences, enhance the overall customer experience, and maintain a competitive edge in the market.

Failure to prioritize the customer's voice can lead to detrimental effects on a company's profitability. Neglecting to meet customer expectations can result in reduced satisfaction, decreased customer retention, and negative word-of-mouth, ultimately leading to financial losses. Conversely, companies that actively embrace and respond to customer feedback have the opportunity to foster customer loyalty, increase retention rates, and drive revenue growth.

Hence, in the constantly expanding e-commerce landscape, acknowledging the importance of customer opinions and experiences is crucial for business success. By leveraging the customer's voice, companies can make informed.

1.3 OBJECTIVE OF PROJECT

Safeguarding the company's financial well-being involves enhancing customer satisfaction and placing significant value on their feedback. By prioritizing customer contentment and actively valuing their input, the company can mitigate the risk of financial losses. This entails striving to meet customer expectations, addressing their concerns promptly, and continuously improving products, services, and overall customer experience.

What benefits will these solutions bring to the company?

By prioritizing customer satisfaction and valuing their feedback, companies can safeguard themselves from financial losses. When customers are satisfied with their overall experience, they are more likely to continue purchasing products or services from the company. This leads to increased customer loyalty and higher retention rates, which, in turn, positively impact revenue generation.

Actively seeking and considering customer feedback allows companies to identify areas for improvement and address potential issues promptly. By addressing customer concerns, companies can enhance their offerings, refine their strategies, and provide a more tailored and satisfactory experience to their customers. This proactive approach helps prevent customer dissatisfaction and potential negative word-of-mouth, mitigating the risk of losing customers and revenue.

Additionally, giving importance to customer feedback helps companies stay in tune with evolving customer preferences, market trends, and industry dynamics. This enables them to adapt their products, services, and business strategies accordingly, ensuring their offerings remain relevant and competitive in the marketplace. By continuously meeting and exceeding customer expectations, companies establish a positive brand reputation, attracting new customers and fostering long-term business growth.

Moreover, valuing customer feedback fosters a sense of customer-centricity within the organization. When employees recognize the importance of customer satisfaction and understand the impact of their actions on the overall customer experience, they become more attentive, responsive, and committed to delivering exceptional service. This customer-centric culture permeates the entire company, leading to improved customer interactions, higher customer satisfaction levels, and ultimately, a more profitable and sustainable business.

1.5 DATA SOURCE

Mass opinion plays an important role in the upgradation of the quality of the product and service improvement of the deliverables [3]. Review sites, blogs, data, and micro blogs gives a good knowledge regarding current opinion of stuff.

Review sites:

Various e-commerce sites such www.amazon.com, ebay.com, www.flipkart.com are available where lot of reviews are available for the product given by customers. Other sites such as www.yelp.com is a restaurant review site, www.nykaa.com is cosmetic product site where large number of reviews regarding various products are available. Hence dataset can be extracted from these review sites.

Blogs:

In recent timings blogging has become very popular as blogger record their daily life event and update on blogs where they even review products and talk about various topics. Hence blogs can be used as a source of data collection for opinion analysis.

Dataset:

Most of the paper classification is done using movie reviews data which is available as dataset (http://www.cs.cornell.edu/People/pabo/movie-review-data). http://www.cs.uic.edu/liub/FBS/CustomerReviewData.zi p is also a review dataset available. Another dataset which is available online is multi-domain sentiment (MDS) dataset [3].

Micro-blogging:

Tweets are status messages posted on the Twitter microblogging service. Users often share their opinions about various topics through tweets. Sentiment analysis is also executed by using Twitter messages.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1. INTRODUCTION

In this paper a summarized review of multiple work related to sentiment analysis is considered. Various approaches have been seen using different data set, algorithms, and pre-processing techniques to enhance the efficiency of sentiment analysis. Most basic step of sentiment analysis is discrimination between subjective text and objective text. Subjective text is one that contain some opinion whereas objective text is just a fact it carries no emotion.

Example:

Objective – Apple has launched new iPhone on 7th of September (it's a fact)

Subjective – iPhone 13 is a very beautiful phone (opinion).

Sentiment analysis is performed on the second example which is subjective by nature as it represents opinion regarding phone. Opinion can be categorized into various classes most analyzed in our survey are neutral, positive, and negative.

Sentiment Analysis is the part of Natural Language Processing (NLP) where we try to train the machine in such a way that it generates the ability to define the overall opinion about certain context such as negative, neutral, or positive. Data is taken and various pre-processing steps are applied and tagging of data is done to define its orientation then this minimalized data is converted into vector space as machine understand numbers not text using sentiment score or weightage to each word or frequency count methods. Then various machine learning algorithms are applied, and results are evaluated.

Understanding emotion behind the text is the complex process many emotions such as sarcasm, irony are hard to determine by machine for example word jaguar whether it's a car or a animal is a complex process to identify and lot of new methods such as deep learning, convolutional neural network, genetic algorithms and combination of old one are coming into scene to deal with implicit sentences such as 'samsung phones are any given day better than oppo phone' this statement is a negative statement but in a implicit way hence hard to determine its polarity.

2.2. LITERATURE REVIEW

Xing Fang and Justin Zhan has done sentiment classification on sentence level and review level using various steps such as sentiment phrase identification that is negative of phrases, score computation, feature vector generation, and polarity categorization [4]. POS tagger is the part of speech tagging of the word in the speech category such as noun, pronoun, verb, adverb, adjectives, conjunction, preposition and many other [1]. It helps in having better understanding of relation between words in a sentence. POS tagging of words is done to extract subjective content from the text. Subjectivity of content is words that signifies the sentiment or represent some opinion. Sentiment score computation is done using formula:

$$SS(t) = \frac{\sum_{i=1}^{5} i \times \gamma_{5,i} \times Occurrence_i(t)}{\sum_{i=1}^{5} \gamma_{5,i} \times Occurrence_i(t)}$$

Where t is number of token and occurrence of t is number of times that token appear in the ith review. Here balancing is done using formula:

$$\gamma_{5,i} = \frac{|5 - star|}{|i\text{-}star|}$$

Here no tagging is required as we already have star tagging on the scale of 1 to 5 where value of score below 3 is negative, value above 3 is positive and value of score 3 is neutral. Sentence level categorization for tagging or label we use bag-of-words model which contain list of positive words and negative words and score is computed based on number of positive or negative word count in a statement and whose count is greater is the sentiment of statement.

Feature vector is developed on four features such as two hash table, tag, and averaged sentiment score. Where hash table represent number of bits to represent number of words token and phrase token. Then F1-score is computed using:

$$F1_{avg} = \frac{\sum_{i=1}^{n} \frac{2 \times P_i \times R_i}{P_i + R_i}}{n}$$

F1-score for sentence level is better as compared to review level. In general, both showed a promising result, but as neutral class is taken into consideration in review level its score is low as compared to sentence level.

Transformation and filtering

Performance is increased in sentiment analysis by pre-processing the data using various techniques [5]. Text preprocessing and then evaluating its performance has shown good results in work published by Haddi. Sentiment Analysis is done in three steps: features are extracted using transformation and filtering using chi square statistics, SVM classifier on feature matrix is applied computed using feature weighing methods feature frequency, feature presence, Term Frequency Inverse Document Frequency, and then performance is evaluated. These weights determine the importance of the feature by attaching weights.

Transformation can be done in many ways such as:

Tokenization: is the concept of splitting the data into smaller units call tokens where they can be classified as n grams where n=1, 2, 3... as n=1 means unigram single word as a token ‘nice’, ‘good’ whereas n=2 means bigram which is combination if two words as a token ‘not appropriate’, ’poor build’. Most used is whitespace tokenization where words are tokenized based on white space [2].

Lemmatization: Removal of inflection from the token and convert it into its original form such as loving into love, failing into fail, and passing into pass.

Stop-word removal: Words which commonly used in formation of English statement are stop words such as a, an, the, from, where. Stop words are of no use as they contain no value with respect to sentiment identification.

Removing Punctuations and Whitespaces: This helps in reducing the size of training dataset like commas, full stops and all which have no value addition.

Filtering is done by chi squared statistics to check the dependency between the word and category it is present in document. Hence irrelevant feature is removed from dataset using 95% level of significance.

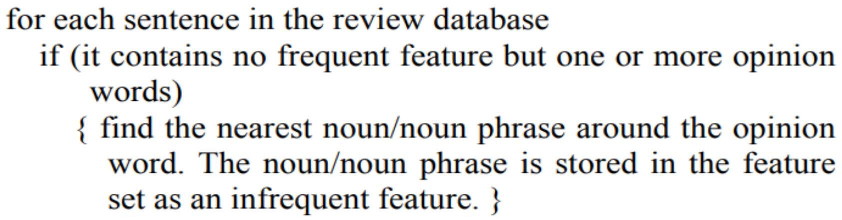
Working with pre-processed data using SVM has shown a great performance as compared to unprocessed data whereas using chi square statics has increased the quality of classifier even more in all three matrices FF, TF-IDF, FP.

<u>Association mining</u>

Feature based sentiment analysis can also be done such that it helps us to understand the opinion of the features of the product such as size, weight, build quality extra [6]. This is basically sentiment analysis on aspect level. Three step processes for mining of customer review: Finding out which product features customers have expressed an opinion about, finding out the number of positive or negative reviews for each feature and result evaluation. Using POS tagger words in sentence are tagged. Using association mining frequent features are extracted. Features which are of no use are removed using compactness pruning and redundancy pruning.

Compactness pruning is removal of unlikely feature by calculating distance between words in the feature phrase such as “I had searched for a digital pen for 3 months.” “This is the best digital pen in the market” “The pen does not have a sharp digital nib” here digital pen appeared together in two sentences hence it is a compact feature used frequently. Redundancy pruning uses p-support (pure support) such that for feature which is either noun or noun phrase no superset must exist for example pen or digital is redundant if digital pen exist hence, they should be pruned.

Here they used a small list of seed adjectives that are tagged positively or negatively. This list grows iteratively using WordNet. This is how semantic orientation of word is predicted. In addition to frequent feature identification infrequent feature identification is also taken in considering such as:



```
for each sentence in the review database
   if (it contains no frequent feature but one or more opinion
       words)
       { find the nearest noun/noun phrase around the opinion
         word. The noun/noun phrase is stored in the feature
         set as an infrequent feature. }
```

Fig 1: Infrequent feature identification algorithm

While predicting orientation when feature review is straight negative or positive then no issue is there but if multiple feature review is given in a single statement and orientation comes out to be neutral that it will result in the loss of information for that we use distance such that opinion words close to that feature represent the opinion for that feature. For example, ‘though touch screen of phone is amazing, but camera is not up to the mark’ here 1 positive word and 1 negative word results in the neutral hence if we look for a particular feature then we can classify as for touch screen its positive and for camera its negative.

Accuracy improves as we use compactness pruning and p-support as compared to only using association mining. Average accuracy for predicting sentence orientation is 84%. This shows method of analyzing reviews using seed list of adjectives is effective. Limitation of paper is it didn’t consider the pronoun such as ‘it’ for example ‘it has a very good touch pad’ here this approach will fail to identify it. Apart from this it gives promising result on adjectives, verb and all is not taken into consideration.

<u>Feature extraction using TF-IDF</u>

Impact of feature extraction on Sentiment Analysis using TF-IDF and n gram is done using SS Tweet dataset [7]. TF-IDF stands for Term Frequency – Inverse Document Frequency it defines the importance of the word by calculating. The weightage of the term using IDF is calculated using formula:

$$W_{x,y} = tf_{x,y^{*}}\; log\left(\frac{N}{df_{x}}\right)$$

$W_{x,y}$= Word x within document y

$tf_{x,y}$= frequency of x in y

df_{x}= number of documents containing x

N=total number of documents

TF-IDF transform the textual information into a Vector form. Let us consider a document having 100 words and the word keyboard appears 5 times that document then the term frequency will be 5/100=0.05 and suppose there are 25000 documents, and 250 documents contain the term keyboard than IDF = 25000/250=100 and TF-IDF is equal to 0.05*100=5. In n gram feature extraction value of n can be 1,2,3…. Where n=1 is unigram and n=2 is bigram and so on such statement ‘laptop build quality is very good’ then for unigram single word is considered in a sentence {‘laptop’, ‘build’, ’quality’, ’is’, ’very’, ’good} whereas in bigram group of words is considered such as {‘laptop build’, ‘build quality’, ‘quality is’, ‘is very’, ’very good’}.

Using various classification algorithms, the conclusion is that TF-IDF features performs better than n gram features by 3-4 % and among various classifiers logistic regression showed the great result.

Chaining classifier and skewness removal

Chaining classifier such that two or classifier are chained one after the other for better performance used in paper [8] where first classifier is used to classify text in three categories opinion oriented such as polar, unnecessary, and neutral. Then in second stage data under polar categorization is used and further classified as positive and negative.

Removing skewness from the dataset can also enhance the performance. That is balancing the unbalanced training dataset using sampling process such as under sampling and over sampling.

Under sampling - eliminating the objects from the majority class.

Over sampling - increasing the objects in the minority class.

The dataset used for first classification irrelevant|polar|neutral discrimination has shown that accuracy has been improved to an extent as we have reduced the degree of skewness [8]. Average accuracy of SMO has increased by 4 to 5% using under sampling and over sampling process. In second classifier SVM gives the consistent result, but the overall accuracy is less as compared to first classifier.

Using neural network

Convolution neural network also a feed forward network requires numerical training vector along with training labels to train network which contain many hidden layers [10] has shown accuracy of 74.15% when performed on dataset used in sentiment analysis competition in the University. But showed uncertain result when used with tweets collected from twitter API [9]. Using Morphological Sentence Pattern (MSP) has shown an improvement in accuracy by 10-11% where neural network is trained on aspect-expression pair [9].

Other approaches

The Recursive Neural Network (RNN), developed by Socher, is a well-known deep learning system. Socher represented movie reviews from the rottentomatoes.com website using properly labelled parse trees. In order to represent the phrase at the root node, recursive neural models compute parent node vectors by combining the vectors of their two child nodes. This is the most basic variation of Socher's Recursive Neural Network. Socher's Recursive Neural Tensor Network (RNTN) greatly enhanced this method. Utilizing a tensor-based composition function for all nodes is the main concept underlying RNTN. They also included a parameter that takes the spacing between words in a phrase into consideration. However, properly labelled parse trees, which might be labor-intensive to collect, are needed for both RNN and RNTN in order to train the neural network. Instead of completely labelled parse trees, Convolutional Neural Networks (CNNs) just need sentence-level labels. Additionally, CNNs are simpler to train since they contain fewer connections and parameters.

Convolutional Neural Networks have shown to be incredibly successful in completing computer vision-related tasks. Convolutional and pooling layers integrate the outputs of neuron clusters to create CNNs. However, the width and depth of CNNs may rise when tackling increasingly complicated issues, resulting in computer resource constraints. Training huge deep convolutional neural networks has become possible due to GPU computing's efficiency. Convolutional Neural Networks have shown to be incredibly successful in completing computer vision-related tasks. Convolutional and pooling layers integrate the outputs of neuron clusters to create CNNs. However, the width and depth of CNNs may rise when tackling increasingly complicated issues, resulting in computer resource constraints. Training huge deep convolutional neural networks has become possible due to GPU computing's efficiency.

There are several difficulties in evaluating the sentiment of user-generated material on social media networks. Diverse methodologies have been developed by researchers, many of which make use of deep learning algorithms. Sentiment analysis tasks have been investigated using Recursive Neural Networks (RNNs), Recursive Neural Tensor Networks (RNTNs), and Convolutional Neural Networks (CNNs). Due to their efficiency in computer vision tasks and their capacity to do sentiment analysis with less parameters, CNNs in particular have demonstrated promise. These developments in deep learning methods have made it possible for sentiment analysis models to do complex tasks.

The link between quantitative and qualitative measures of financial performance has led to the development of sentiment analysis in finance, a prominent field of research. Loughran and McDonald's groundbreaking study has shown that word lists made for other professions frequently misclassify phrases that are frequently used in financial journals. Loughran and McDonald created an expert-annotated vocabulary for positive, negative, and neutral phrases in finance to solve this problem . The authors of proposed a Twitter-specific vocabulary that, when used with the DAN2 machine learning technique, outperforms the support vector machine (SVM) approach using the same Twitter-specific lexicon. This is in the area of sentiment classification. Sentiment has been extracted from datasets comprising tweets or news items using machine learning techniques. To demonstrate the SVM classifier's better performance in comparison to Decision Trees and Naive Bayes classifiers, the authors of use a variety of machine learning binary classifiers to extract sentiment from StockTwits tweets. In order to predict a real-valued sentiment score in microblogs and news headlines, Atzeni et al. investigate the efficiency of regression models and use statistical and semantic feature extraction approaches. They emphasise that applying semantic approaches improves classification precision.

For sentiment extraction, researchers have used a combination of lexicon-based approaches and machine learning models. As opposed to depending on a single model for sentiment extraction, it has been demonstrated that using such combinations produces superior outcomes . Standard machine learning methods have trouble extracting complicated attributes and capturing the word order of a sentence. As a result, sentiment extraction in the field of finance has been successfully accomplished using deep-learning models such as recurrent neural networks (RNNs) , convolutional neural networks , and attention mechanisms improves classification precision. The remarkable success of deep-learning approaches in natural language processing (NLP) can be attributed to advancements in text representation methods such as word and sentence encoders. These techniques convert phrases or sentences into vector representations so that neural networks can use them as input. For sentiment extraction, these representations keep the semantic information that is inherent in words and phrases. Sentiment extraction from financial news and texts has been considerably enhanced by recent developments in NLP, deep learning, and transfer learning. ULMFiT and other inductive transfer-learning techniques are used by Yang et al to demonstrate superior sentiment categorization over conventional transfer-learning techniques. The RoBERTa model is demonstrated to outperform dictionary-based models in the evaluation of contemporary NLP transformers, such as BERT and RoBERTa, in sentiment analysis.

In summary, the link between quantitative and qualitative measures of financial performance makes sentiment analysis in finance an interesting study topic. The study of Loughran and McDonald emphasises how critical it is to develop specialised languages for sentiment analysis in finance. SVM classifiers and other machine learning tools have been used to extract sentiment from tweets and financial documents, with substantial progress being made thanks to deep learning models and text representation approaches . Sentiment analysis of financial news and texts has recently improved thanks to advances in NLP, deep learning, and transfer learning. Transformer models like BERT and RoBERTa have shown to perform better in this area.

The difficult process of sentiment analysis is gleaning useful information from vast amounts of textual data. To analyse data from social media networks, researchers use a variety of techniques, including lexicon-based and machine learning-based approaches. Due to their capacity to recognise characteristics from large datasets, machine learning approaches like Support Vector Machines and Deep Neural Networks have shown to be efficient in sentiment analysis. However, these models must be trained using a lot of labelled data, which may be time- and resource-consuming. Due to their feature extraction skills and capacity to recognise local patterns, Convolutional Neural Networks (CNNs) and Transformers like BERT and RoBERTa are frequently employed with word embeddings in the field of text processing to categorise and cluster data.

SNo.	Year	Title	Dataset	Approach	Advantage	Limitations
1.	2015	Sentiment analysis by Xing Fang & Justin Zhan.	Amazon	Polarity categorization on sentence level and review level	Results are goods with F1 score of 0.81 in sentence level and 0.73 in review level	Review classification to specific star scale is difficult and it does not work well for implicit sentiment.
2.	2013	Impact of text pre-processing by Emma Haddi, Xiaohui Liu, Yong Sh.	Movie	Text preprocessing and chi square statistics for filtering using with SVM	Performance of classifier is increased	Shows better result only when using unigram with classifications.
3.	2004	Summarizing and Mining customer reviews by Bing Liu and Minqing Hu.	-	Sentiment orientation of sentence on feature of product using compactness pruning and p-support.	Average accuracy for predicting sentence orientation is 84% using WordNet	pronoun in sentence (it) not considered and only adjectives used for orientation
4.	2019	Impact of feature extraction on	Sentiment Strength	Analysis of text is done	When compared	Implicit features are

		sentiment analysis by Ravinder Ahuja, Aakarsha Chuga, Shruti Kohli.	Tweet dataset	using two Feature extraction technique TF-IDF and n gram	to N-Gram features, TF-IDF features produce better results (3-4%)	not taken into consideration such as irony and sarcasm.
5.	2012	Sentiment analysis and opinion mining Nadarajah Prasath, AShehan Perera, and Balakrishnan Gokulakrishnan.	Twitter	Removing skewness and using chained classifier for sentiment analysis	Performance of SMO has increased from 77.1% to 81.9% for first classifier.	Sarcasm, international expression can also be explored.
6.	2018	Sentiment Classification of Tweets with Non-Language Features by Jothi Gb and Akilandeswari Ja.	Twitter	Sentiment Analysis of text having Non-Feature Language	Performance of classifier is better when NFL is considered	Model can use suitable statistical techniques to examine the classification performance
7.	2015	Neural Networks for Sentiment Analysis on Twitter by Yanqing Zhang and Brett Duncan	Tweets were collected from the University competition	The neural network used is the feedforward pattern network	The average accuracy was 74.15%	Memory issue and giving 50% accuracy when used on twitter API dataset.

8.	2018	Sentiment Analysis using Neural Networks: A New Approach by Shiv Dhar, Suyog Pednekar and Kishan Borad	Twitter	A combination of MSP model and Convolutional Neural Network (CNN)	Accuracy is improved from 10-11% by using MSP	various other combinations of part-of speech tags and other grammatical structures can also be taken into consideration

Tabular representation of literature review

2.3. OVERVIEW

This section will describe the planned approach for doing histopathological image detection for cancer, which is implemented on a deep convolutional neural network model. In which we designed a deep convolutional neural network as from scratch.

Classification is done based on polarity categorization nonpolar | negative | positive, which is frequently studied in sentiment analysis. An analysis of sentiment is often conducted at one of the following levels: the document level | the attribute level | sentence level. You can conduct sentiment analysis using two approaches, namely machine learning and lexicon-based analysis.

2.3.1. Machine Learning Approach:

Machine learning is a field of Artificial Intelligence. Model is trained such that it develops the capacity to learn from data.

There are two types of Machine Learning Approach: supervised and unsupervised.

In supervised learning model is trained using labeled data. Data is divided into training and testing data set [3].

Naïve Bayes: It is a probabilistic algorithm that uses probability theory along with Bayes' Theorem to predict the tag mentioned in the text. It is probabilistic, it output the tag with the highest probability for each tag based on the probability of each tag appearing in each text. Basically, it determines the probability of the tag in the sentence or text for example in case of polarity categorization our tags can be positive | negative | neutral. And then calculate the probability of each word using bayes formula:

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

Here A is 'The movie was very bad' and B can be positive or negative. We calculate the probability for both the tags and consider the one whose value is greater.
P (the movie was very bad | positive)
and
P (the movie was very bad | negative)

Here, we assume every word is independent of each other. Though we cannot use text directly to calculate probability we need numbers which we get in terms of frequency used for each word that is the count of its appearance. Class whose probability is higher is the correct output.

SVM: it is a non-probabilistic supervised machine learning which uses two group classifications. SVM takes data point as an input and plot them on a multidimensional space and output the hyperplane which separates the two groups. This hyperplane is the decision boundary which helps in deciding where our tag belong to. The best hyperplane is one whose distance from the tags closest to the hyperplane is maximum. When our data is linearly separable then our hyperplane is a straight line in two-dimensional space but when our data is nonlinearly separable, we need to add more dimensions and to make this process less expensive we use kernel function that using dot product approach. In comparison to neural network SVM has better performance and better speed for limited sample data.

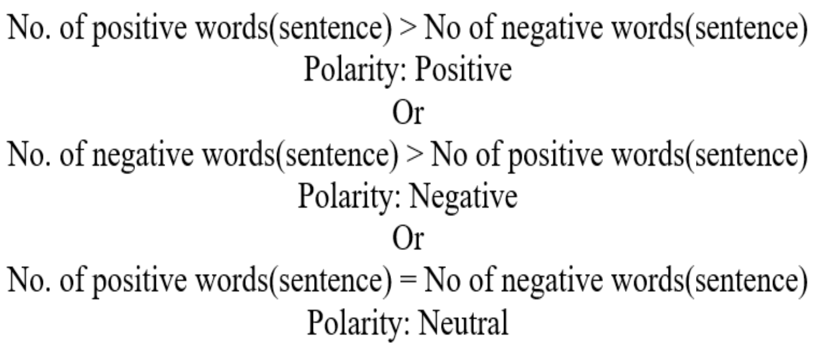
The unsupervised learning techniques determine the labels for each data point based on internal differences between them. Basically, unlabeled data is fed into model where desired outcome is not known and inferences are made based on circumstantial evidence without any guidance. Most used unsupervised mechanism is clustering which cluster the same type of data.

K-Means is the unsupervised machine learning algorithm which classify the dataset into K clusters. Here cluster has the centroid where centroid is the center of each cluster. Initially these centroids are present randomly on the vector space and then move at the center of the points which are close to centroid. This work iteratively where after every iteration centroid distance is re-calculates from its nearest point and centroid move to the center again. K Means is a method of finding groups from unstructured data.
In sentiment analysis unsupervised learning is used mostly in defining the semantic orientation of the text. How much the word is lead towards positive | negative.

Deep learning is advanced machine learning [4]. They imitate the way human brain works and are built on multiple number of layers of interconnected neuron allowing many systems to work together simultaneously. In case of complex problem and huge dataset deep learning take the edge over machine learning.

2.3.2. Lexicon Based Approach:

In this approach it uses rules to classify the polarity or opinion of the text. This approach is defined based on dictionaries or two different list of words such negative list containing words such as poor, bad, worst and all whereas positive list of words containing words such good, nice, great and all. Then we keep the count of negative and the positive words present in the text and if number of negative words are less than the number of positive words in the given text than the overall polarity or sentiment or opinion of text is positive.



No. of positive words(sentence) > No of negative words(sentence)
Polarity: Positive
Or
No. of negative words(sentence) > No of positive words(sentence)
Polarity: Negative
Or
No. of positive words(sentence) = No of negative words(sentence)
Polarity: Neutral

It’s a very simple approach and further complexity of statement can be managed by creating more rules but adding more rules affect the whole system and require regular updating.

Hybrid Approach:

To make them more accurate, hybrid approach is used which is done by combining machine learning approach and rule-based approach

2.3.3. Deep learning

Deep learning is subset of ML that contains artificial neural network, that are algorithms inspiredby biological and function of the human brain.

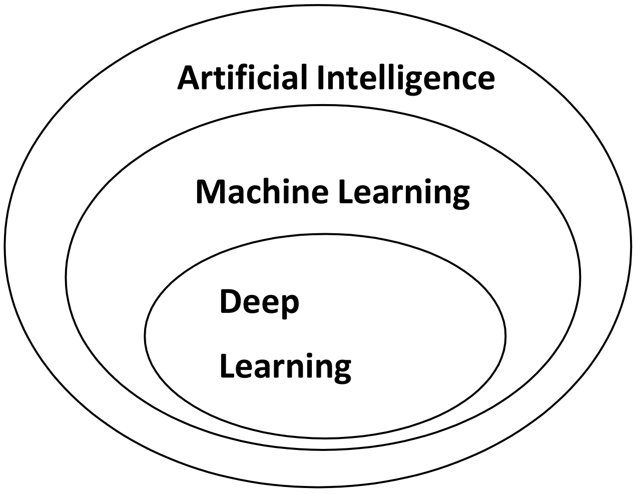


Fig 3: Machine learning, Deep learning and Artificial Intelligence

Deep Neural Networks evaluate data in/ an structured manner to reach comparable results as humans. Deep learning achieves this by putting a algorithms to work with multiple layers known as neural networks.

We may use neural networks to accomplish a wide range of activities, like as, classification, regression and clustering. Based on the patterns between the samples, we may utilize neural networks to categorize / classify unlabeled data. In the classification step, we can train the DNN model on a labeled dataset for categorize the data in this dataset among various categories.

2.3.4. CNN

Convolution neural network is a one type deep learning model. This neural network is feed forwardneural network where signal flow from output of one neuron to input of next layer neuron means there is no feedback.

A CNN is a technique that takes an image is processed and assigns weights to the features in the image so that they can be discriminated. Compared to other classification techniques, CNNs don'trequire that much preprocessing. The filters can be learned by the convolutions themselves. The architecture of CNN is the structure of neurons as a source of inspiration in the human brain.
CNN uses multiplication of an image matrix with a for extracting features and pre-determined characteristics from it. We use a channel to filter the image and get only the predominant importantfeatures. The images are matrices of pixel values, and the filters are commonly 3x3 or 5x5. The

filter is moved across the image with a specified stride, and the values are multiplied and added toprovide a matrix output that is easier to understand.

$$F(x) = f_n(f_{n-1}(\dots f_{\mathrm{i}}(x))$$

Where ‘n’ is number of hidden layer and ‘fi’ is function used in corresponding layer.
In a CNN model there are basically 5 layers.

- Convolution layer

- Activation function layer
- Pooling layer
- Fully connected dense layer
- Predication layer

Convolution layer

In convolution layer it uses filters size n*n and apply convolution operation all over the imageusing these filters and extract the tiny features of the image which we called feature map. Forexample: In nodule detection in lung cancer we use to detect edges, shapes, abnormal cells etc.

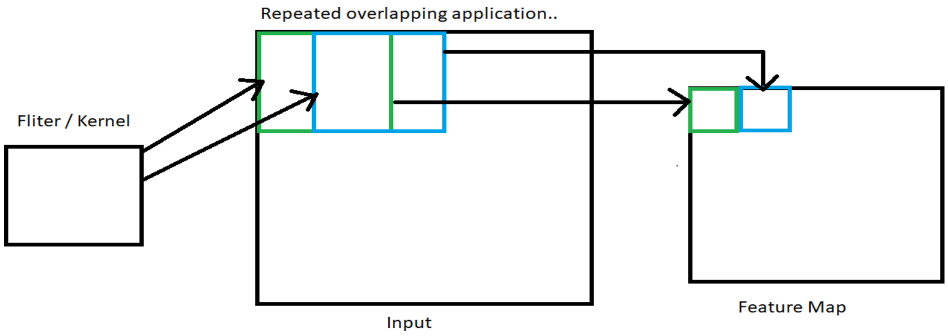


Fig 4: Applying filter in input image

Activation function layer

In this layer we use any non linear activation function to bring non-linearity in our model which speeds up training and faster to compute. It is used to learn and understand complexpattern in our data as well as to prevent the numbers from aggregating to zero. Mostly weuse RELU (rectified linear units) as a activation function which can be expressed as

$$f(\mathrm{x}) = \max(\mathrm{O}, \mathrm{x})$$

There can be other activation function too like sigmoid, tan h and exponential linear units(ELU) etc.

Linear activation function

The equation for a linear function is y= mx, which is the same as the equation for a straightline.

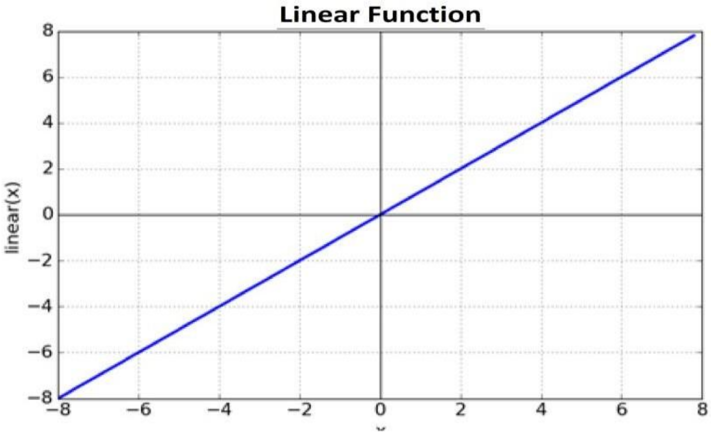


Figure 5: Linear Activation function

Sigmoid Function:

The sigmoid or sigmoid activation function's curve resembles an 'S' shaped curve. Betweenzero and one is the range of the logistic activation function. Because value of the sigmoidfunction is limited between zero and one, the outcome is likely to be one if the value is greater than 0.5 & zero else.

Equation: f(x)= $^{1}/_{(1+e^{-x})}$

Range: 0 to 1

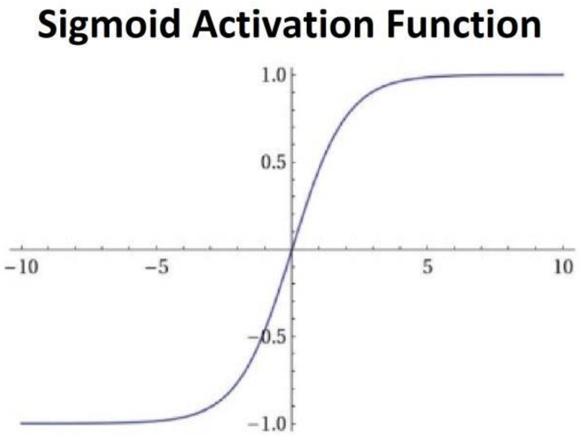


Figure 6: Sigmoid Activation Function

Tanh activation function

Tanh is a hyperbolic tangent function, similar to the logistic sigmoid. The curves of the Tanh and sigmoid activation functions are quite similar, as illustrated in figure 2.13, however Tanh is preferable since the whole function is zero centric.

Equation: f(x)=tanh(x)= $-1/(1+e^{-2x})$

Range: -1 to 1

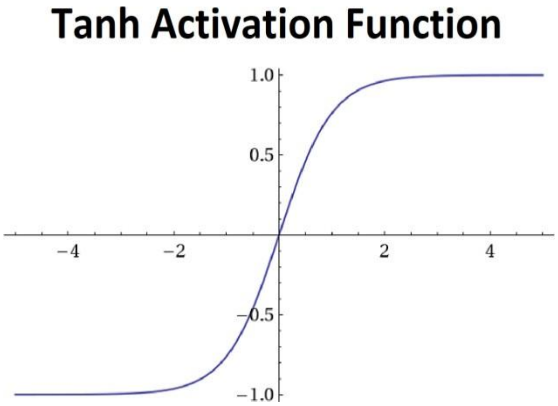


Fig 7: Tanh Activation Function

ReLU activation function

It's most often used activation technique in hidden layers of a deep neural networks. It's themost used activation method in DNN hidden layers. Because the ReLU function is non- linear, we may quickly back transmit errors and trigger multiple layers of neurons. ReLU is less expensive than hyperbolic tangent and sigmoid because it uses fewer complex computations. Because just a few perceptrons are engaged at any given moment, the cnn is sparse and quick to process.

Equation: f(x)
= max (0, x)
Range: [0 to ∞)

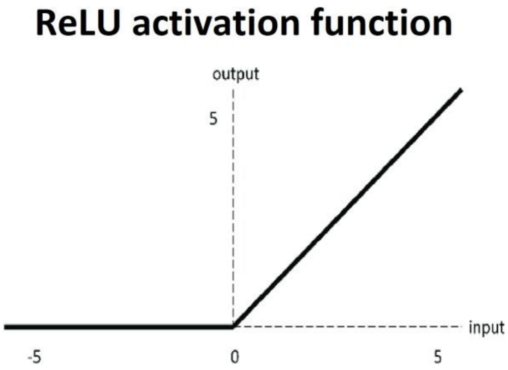


Fig 8: ReLU activation function

Softmax activation function

The softmax function is a function that deals with classification tasks at fully connected layer. When dealing with many classes, this is commonly employed. The softmax functionhas range from zero to one. The softmax function is best used at the output layer of the deep neural network, where we want to use probability to characterize the classification from each input.

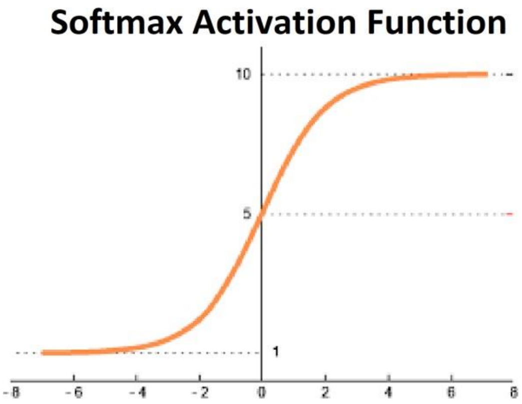


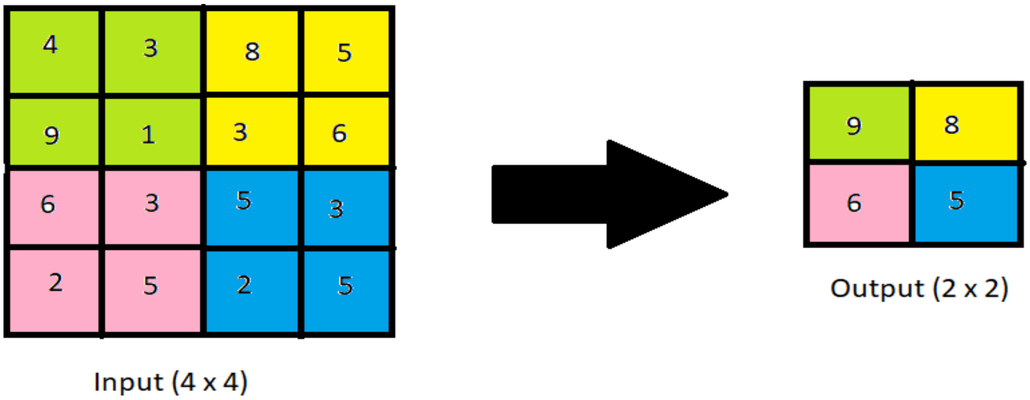
Fig 9: Softmax Activation Function

Pooling layer

This layer actually used to reduce the size or reduce the dimension of image (feature representation). By experiments it was found that Max pooling is used mostly. In Max pooling we use a window of size m*m and take maximum pixel value among all the pixelvalues in window of feature map and slides to stride of ‘k’ and by doing this it cover the whole feature map. There are several forms of pooling, such as maximum pooling, minimum pooling, averagepooling, and so on.

Max Pooling:

A method of determining the highest weight of sectors by pooling them together of a feature map and uses it to construct a down sampled (pooled) map of features is called as Max-Pooling. After a convolutional layer, it's typically used. By employing pooling layers,the size of the convolutional matrix are reduced. As a result, the number of learnable parameters is reduced, as is the network's processing complexity.



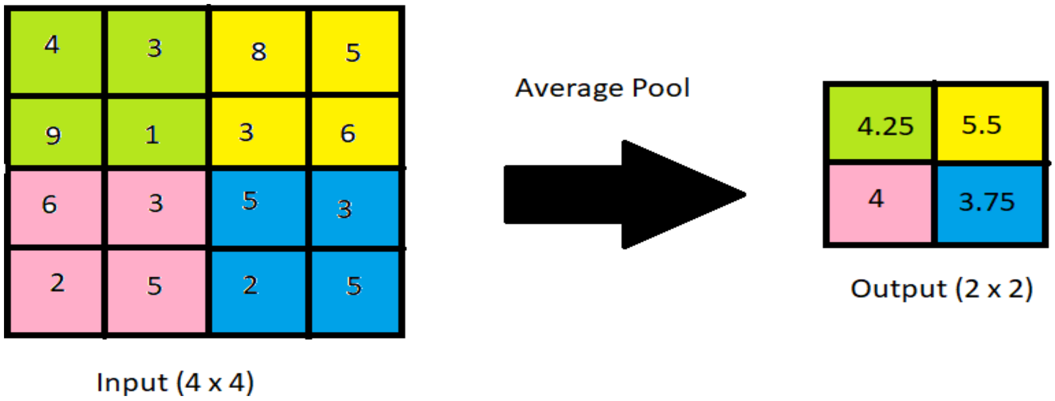
4	3	8	5
9	1	3	6
6	3	5	3
2	5	2	5

9	8
6	5

Fig 10: Applying 2x2 max-pooling on input

Average Pooling:

The average of the values accessible in the region of the feature space covered by the kernelis used in average pooling. The pooling layer creates a connection between the convolutional and fully-connected layers in general.



4	3	8	5
9	1	3	6
6	3	5	3
2	5	2	5

4.25	5.5
4	3.75

Fig 11: Applying 2x2 Average pooling on input

Fully connected dense layer

This is the last layer, where the categorization takes place. Here, we combine our filteredand shrunk images into a single list which is called vector.

In Fully connected dense layer each neuron is connected to every other neuron of nextlayer.

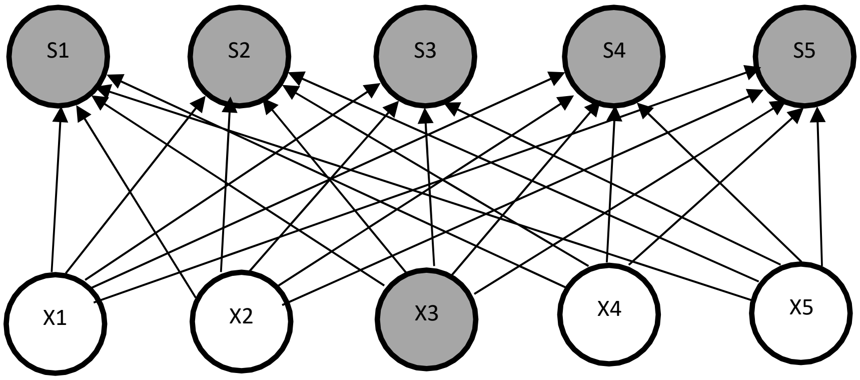


Figure 12: Fully connected dense layer

2.3.5. Dilated Convolution

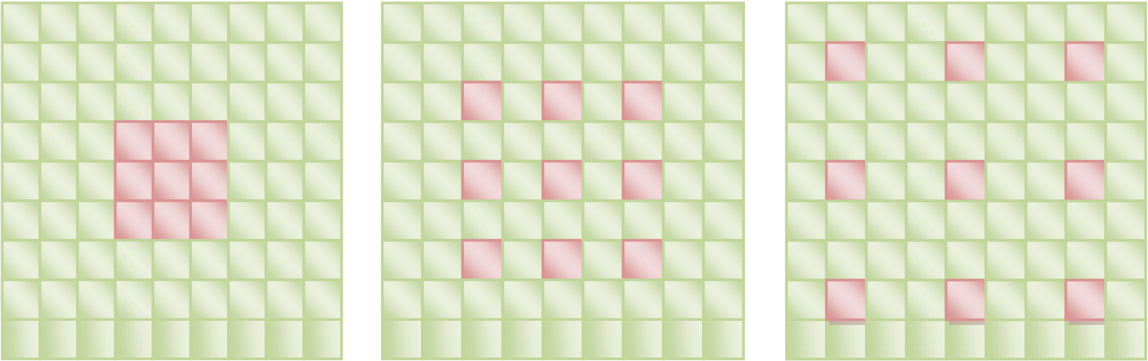
It's a mechanism for expanding a kernel or filter by making holes in between the elements. In layman's words, that's the same as convolution, but now with pixel skipping to cover a broader area of the input. The 'dilation factor' argument specifies just how far the input is stretched. In other terms, the filter skips (dilation factor-1) pixels dependent on the value of dilation rate.

We can get additional information without increasing the number of filter parameters by employing this technique. Dilated convolution allows you to cover a larger region of the input images without pooling. The goal is to extract more details from the output after each convolutionlayer. At the same computing cost, this approach provides a larger field of vision. We calculate the value of a 'dilation factor' by evaluating how much knowledge is collected with each convolution at different 'dilation factor' values.

Dilated Convolution's Benefits:

- A more expansive receptive field is available.
- Effective in terms of computation.
- Memory usage is lower.
- There is no degradation in the produced image's resolution.

- The convolution's structure assists in keeping the information in order.



Kernel size: 3*3

Dilation rate = 1

Kernel size: 3*3

Dilation rate = 2

Kernel size: 3*3

Dilation rate = 3

Figure 15: Dilated Convolution

Graph Neural Network

The semantic, syntactic, temporal, and relational structures of words are captured (via GNN). Graphs have long been an important part of NLP applications such as B. Syntax-based machine translation, knowledge graph-based question answering, and abstract semantic expressions for common-sense tasks. However, with the advent of end-to-end deep learning systems, the use of traditional analysis techniques is diminishing. In reality, because to a lack of fully fresh concepts, there have been several disputes over flattening state-of-the-art NLP systems.

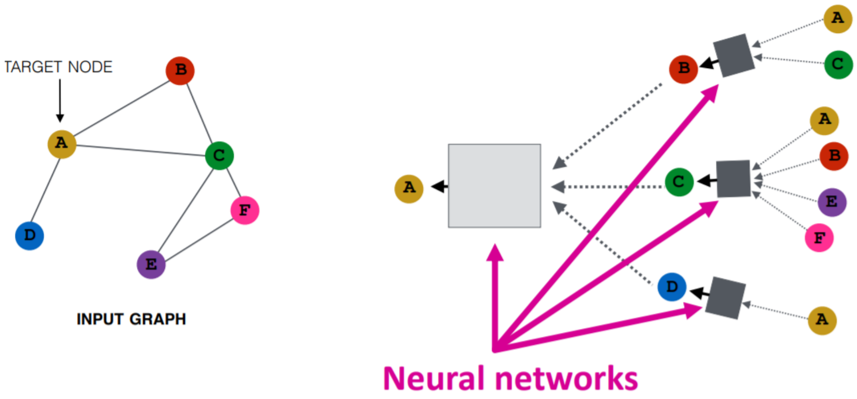


Fig. 16. Graph Neural Network

The graphic depicts the Spacy tagger's parser output. Each node can be defined as a word and each edge can be defined as a dependency analysis tag. Pos tags can be applied to any word.

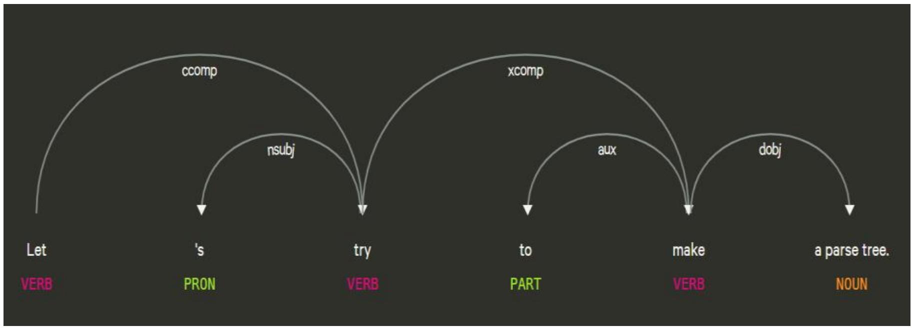


Fig. 17. Hierarchical relation between text

Some claim that strong attention processes can learn syntactic and semantic links automatically. Though, to my knowledge, there is no theoretical study that indicates when attention is inefficient. Consider the problem of aspect-based sentiment analysis (trying to collect brand sentiment that identifies the sentiment of each feature, you can get the sentiment of multiple aspects such as fit, material, delivery, etc.).

2.3.6. Graph Convolutional Neural Network

Over the past decade, there has been a surge in the popularity of neural networks. However, early neural networks were only suitable for regular or Euclidean data, despite real-world data often having a non-Euclidean network topology. The advancement of graph neural networks has been driven by the need to handle irregularities in data structures. In recent years, various types of graph neural networks, including the Graph Convolutional Network (GCN), have been developed. GCN represents one of the fundamental variations of graph neural networks.

Graph neural networks (GNNs) are models that leverage message passing between nodes to capture graph relationships. They extend the capabilities of convolutional neural networks (CNNs) to handle non-Euclidean data. This is achieved through two techniques: a spatial domain filter and a spectral domain filter. These filters extract spatial features from topological graphs. While the spectral domain filter is limited to fixed linked graph processing, the spatial domain filter is more versatile and can be applied to a wider range of applications. scenarios.

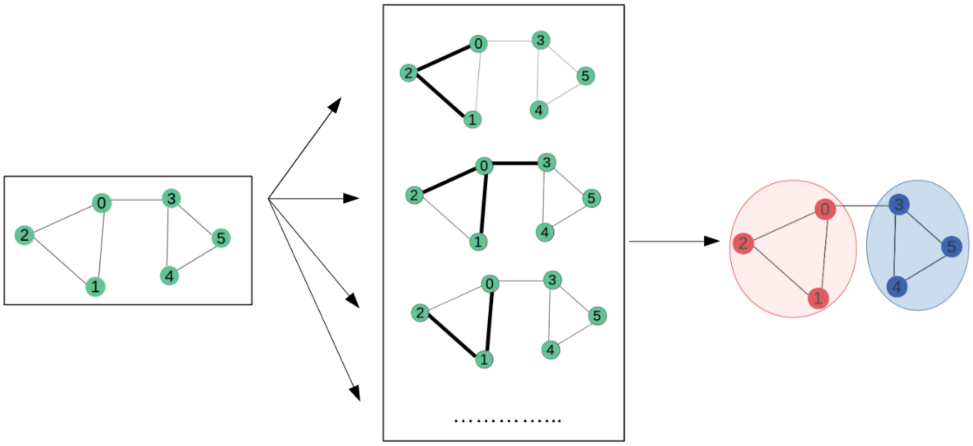
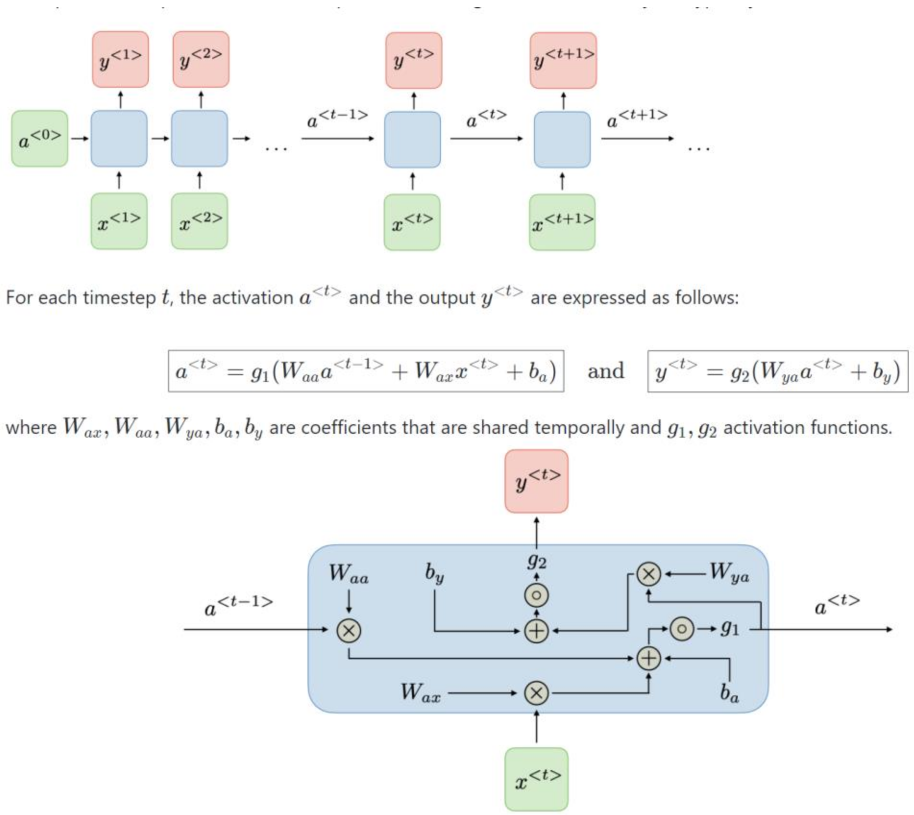


Fig. 18. Convolutions of graph

If you are familiar with the layers of convolutional neural networks, the "convolution" of a GCN is essentially the same. This is the process of applying a set of weights to an input neuron called a filter or kernel. CNN's may learn information from surrounding cells thanks to the filters, which operate as a sliding window over the entire picture. The same filter will be utilized throughout the image inside the same layer, which is known as weight sharing. If you use CNN to identify cats and non-cat photos, use the same filters on the same layer to detect cats' noses and ears.

2.3.7. RNN

In both Google's voice search and Apple's Siri, recurrent neural networks (RNNs), the most recent strategy for consecutive information, are utilized. Because of its inward memory, it is the primary calculation for reviewing its feedback, making it ideal for ML issues like successive information. Artificial neural networks known as repetitive neural networks are frequently utilized in voice recognition and regular language processing. Recurrent neural networks look for patterns in data and use these patterns to guess the next plausible result.



For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$\boxed{a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)} \quad \text{and} \quad \boxed{y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)}$$

where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are coefficients that are shared temporally and g_1, g_2 activation functions.

Fig 20: RNN architecture

2.3.8. Bi-RNN

Bidirectional recurrent neural networks (BRNN) interface two secret layers with restricting result headings. The result layer of this sort of generative profound gaining might get input from both the past (in reverse) and future (ahead) states simultaneously. BRNNs were laid out in 1997 by Schuster and Paliwal to further develop the amount of info data open to the organization. Multi-layer perceptron (MLPs) and temporal delay neural network (TDNNs), for instance, have input information adaptability requirements since they need fixed input information. Standard recurrent neural networks (RNNs) are moreover restricted in that future information data can't be gotten to from the current state. BRNNs, then again, don't require fixed input information. Moreover, future info data is available from the current state.

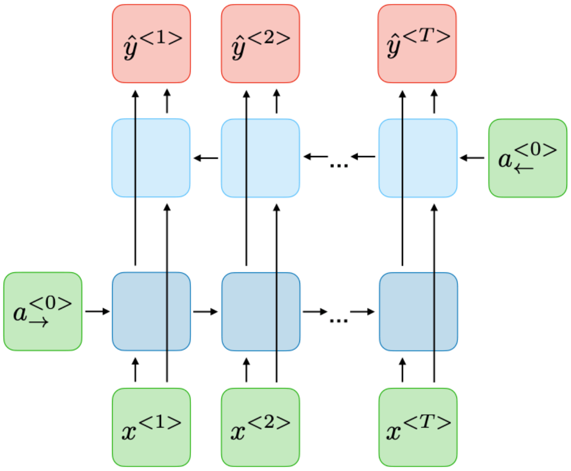
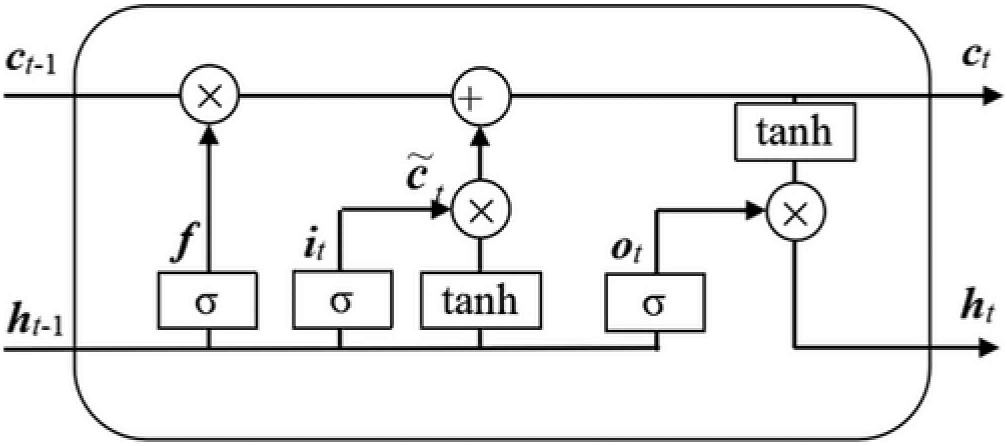


Fig 21: Bi-RNN architecture

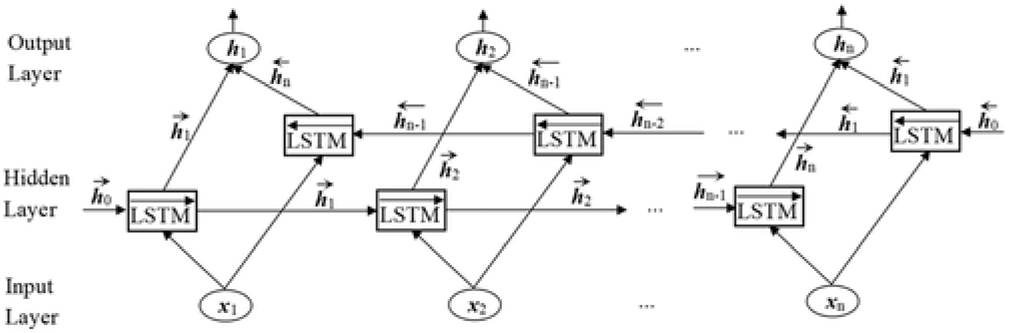
2.3.9. LSTM

The term LSTM alludes to long short-term memory organizations, which are utilized in Deep Learning. In movement guess undertakings, it is a sort of recurrent neural network (RNN) that can learn significant length affiliations. LSTMs utilize an assortment of "entryways" to control how data in an information grouping enters, is put away in, and withdraws the organization. A LSTM regularly comprises of three entryways: a neglect door, an info entryway, and a result entryway. These entryways go about as channels and each has its own brain organization.



2.3.10. BiLSTM

An improvement known as bidirectional long-short term memory (bi-lstm) empowers any cerebrum association to store data both forward and in reverse (future to past) (past to future). Our feedback is bidirectional and heads down two paths, separating a bi-lstm from a typical LSTM.



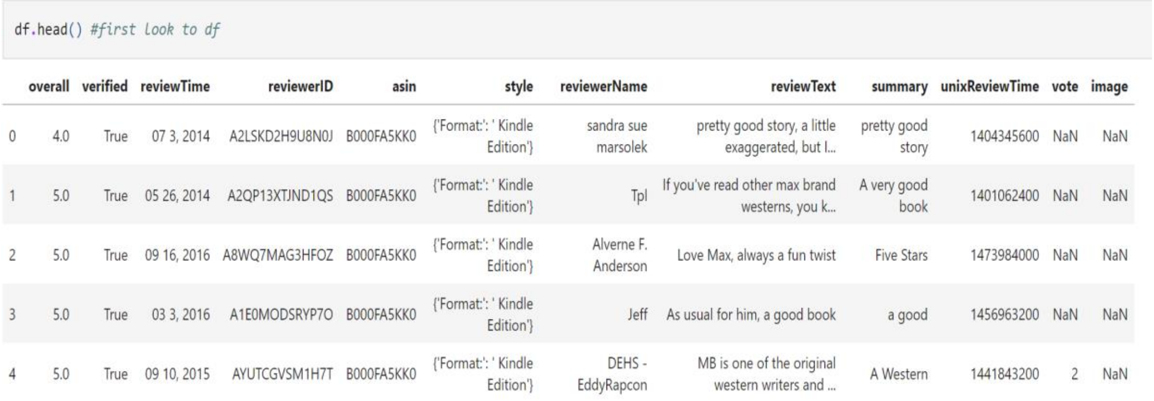
CHAPTER 3

METHODOLOGY

3.1 DATA SET

Data is obtained from UC San Diego Computer Science and Engineering Department academic staff. Metadata and 142.8 million product reviews from May 1996 to July 2014 are included in the dataset. Some academic personnel in the computer science department at UCSD separated the data into different product categories to make downloading easier. Data from the Kindle Store includes 5,722,988 reviews for 493,859 different products. The metadata and the 5-core sample dataset for the Kindle Store (so that each of the remaining users and items has at least 5 reviews). Continuous users contain more information than single reviewers, which is one of the reasons to choose 5-core data. People need to fill out and submit a form in order to access and download metadata.

Dataset contains 2,222,983 rows and 12 columns.



```
df.head() #first look to df
```

	overall	verified	reviewTime	reviewerID	asin	style	reviewerName	reviewText	summary	unixReviewTime	vote	image
0	4.0	True	07 3, 2014	A2LSKD2H9U8N0J	B000FA5KK0	{'Format:': ' Kindle Edition'}	sandra sue marsolek	pretty good story, a little exaggerated, but I...	pretty good story	1404345600	NaN	NaN
1	5.0	True	05 26, 2014	A2QP13XTJND1QS	B000FA5KK0	{'Format:': ' Kindle Edition'}	Tpl	If you've read other max brand westerns, you k...	A very good book	1401062400	NaN	NaN
2	5.0	True	09 16, 2016	A8WQ7MAG3HFOZ	B000FA5KK0	{'Format:': ' Kindle Edition'}	Alverne F. Anderson	Love Max, always a fun twist	Five Stars	1473984000	NaN	NaN
3	5.0	True	03 3, 2016	A1E0MODSRYP7O	B000FA5KK0	{'Format:': ' Kindle Edition'}	Jeff	As usual for him, a good book	a good	1456963200	NaN	NaN
4	5.0	True	09 10, 2015	AYUTCGVSM1H7T	B000FA5KK0	{'Format:': ' Kindle Edition'}	DEHS - EddyRapcon	MB is one of the original western writers and ...	A Western	1441843200	2	NaN

Table 2: Data represented in the tabulation format.

3.2 DATA ANALYSIS

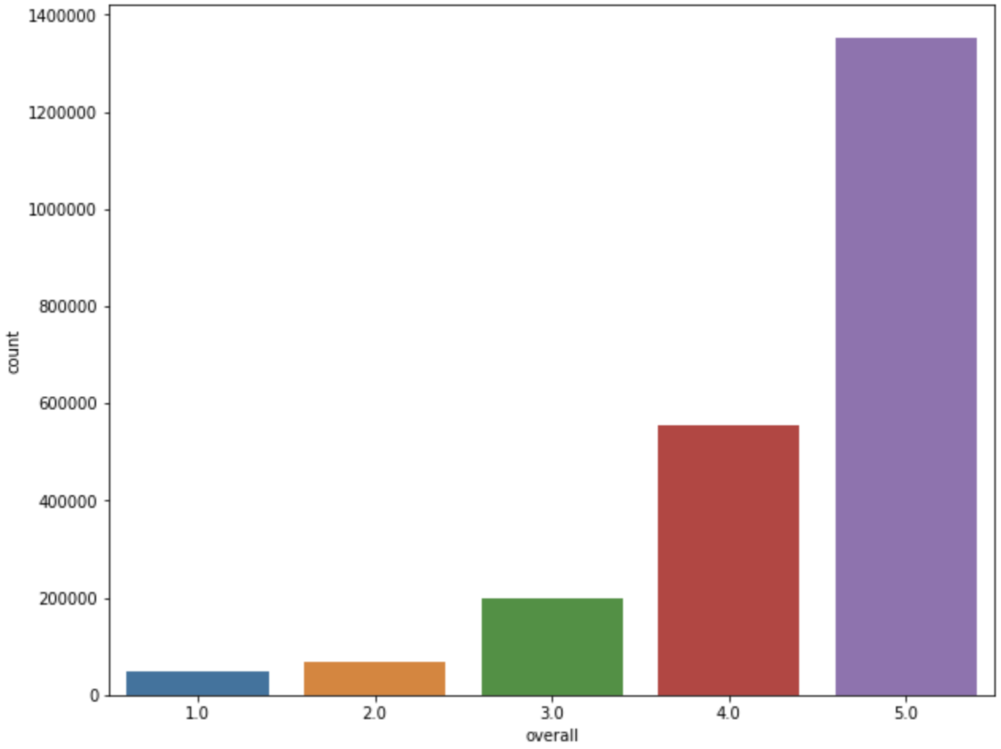


Fig 22: Representing overall rating out of 5 star

The majority of ratings in the sample are four and five star positive reviews. The objective is clearly unbalanced.

According to our data, about 64% of reviews are validated by Amazon. Since this initiative is not focused on detecting fraudulent reviews, I believe that Amazon has not yet verified non-confirmed reviews. Review verification takes time. However, by the time they are verified, I won't make any changes to this column.

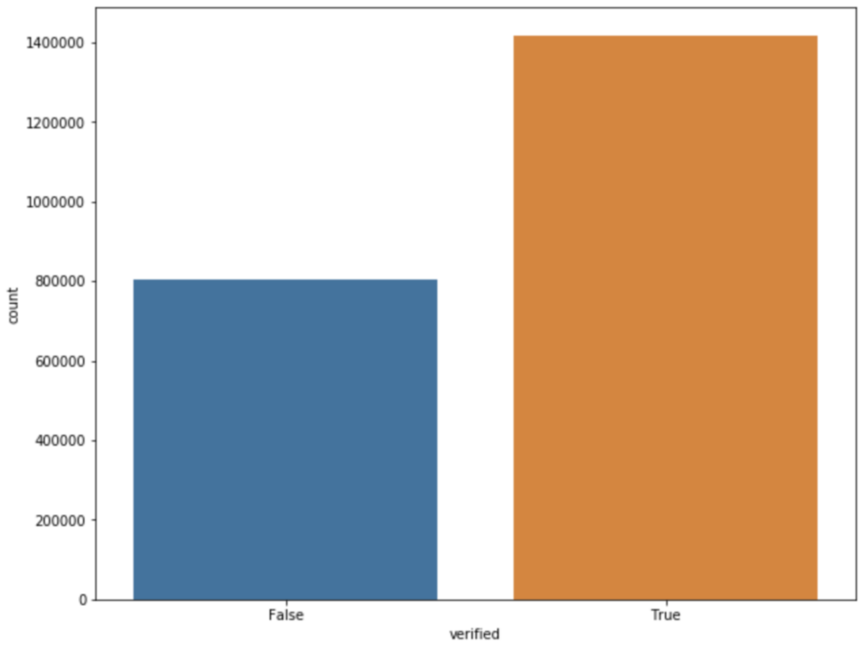


Fig 23: Representing weather the users are verified or not

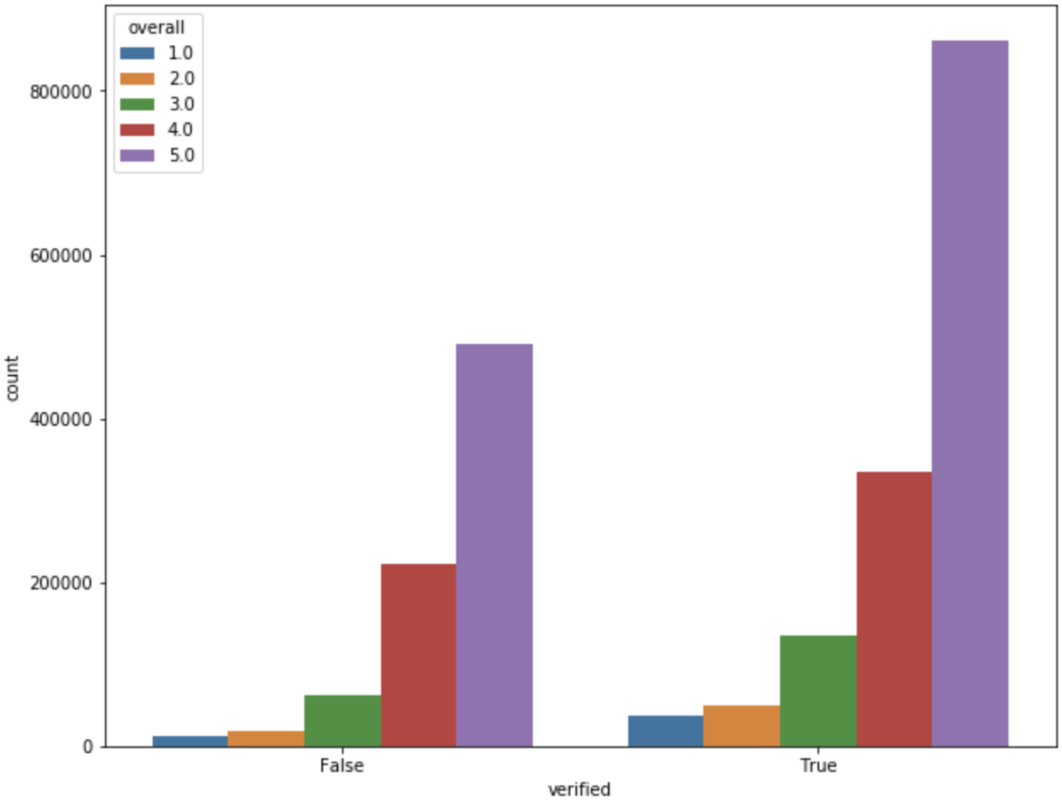


Fig 24: Representing the rating distribution in both verified and unverified reviews is essentially the same.

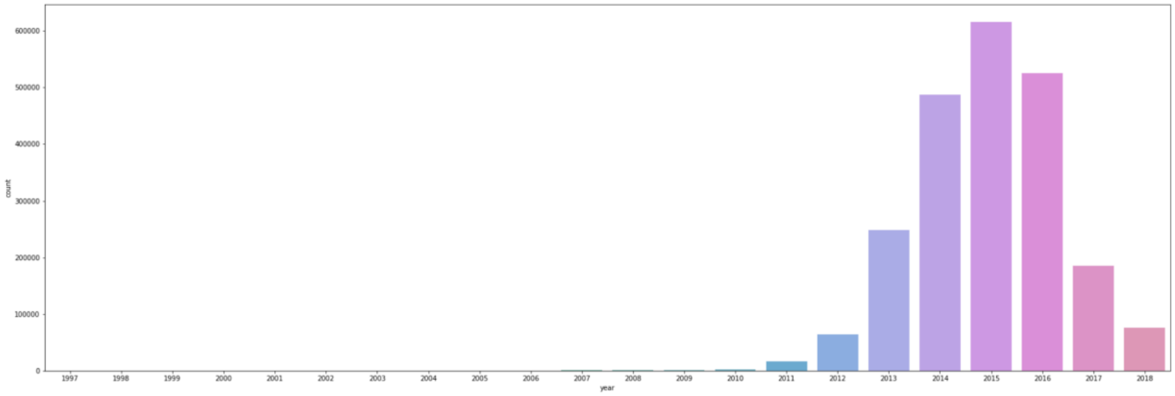


Fig 25: Ratings in a particular year

The 21 years represented by our data are. It is clear from the graph that the majority of the data comes from recent years. It enables us to create future models that are more precise.

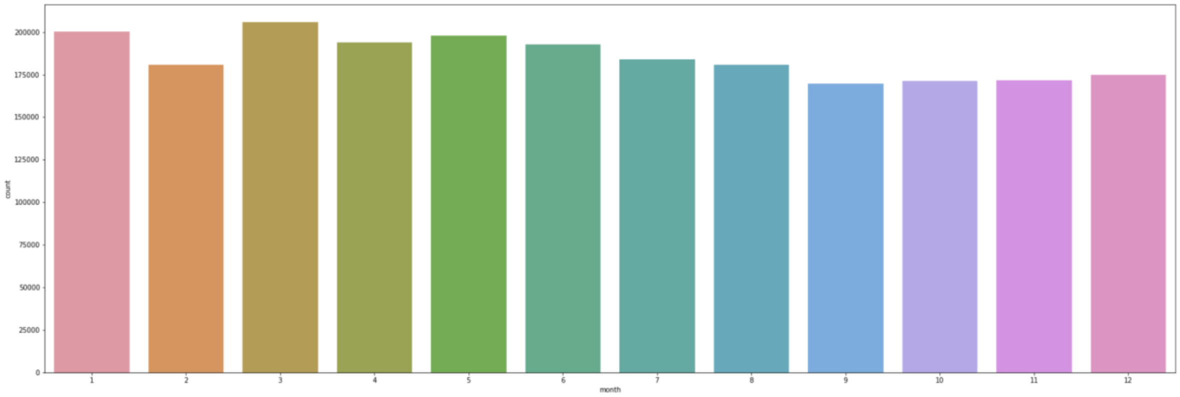


Fig 26: Monthly Ratings

For the most part, our data show similar rating numbers for each month.

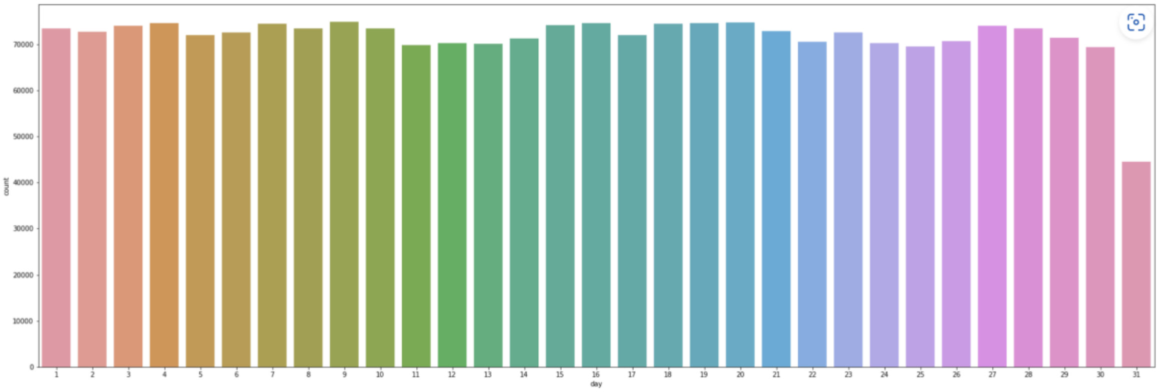


Fig 27: Daily Ratings

Plots show that, with the exception of the 31st, our data distribution is essentially the same for each day. Because there isn't a 31st day in every month. This is a reliable indicator of the accuracy of our data.

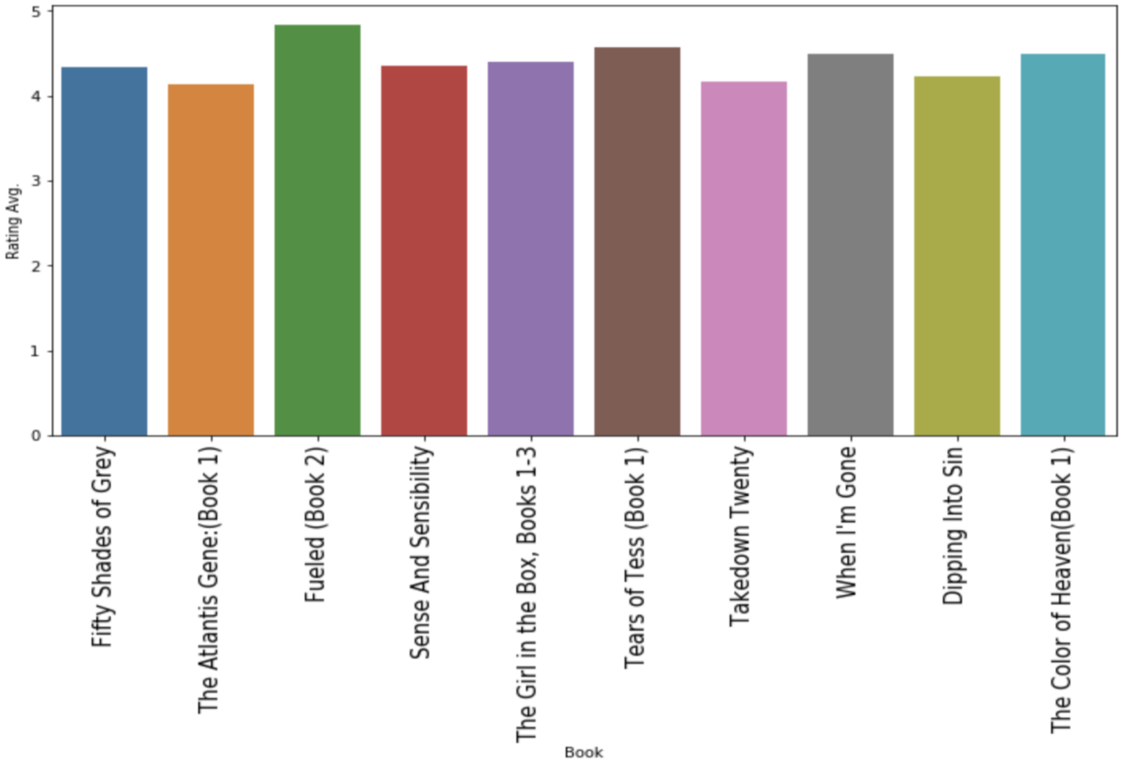


Fig 28: Top 10 highly reviewed books.

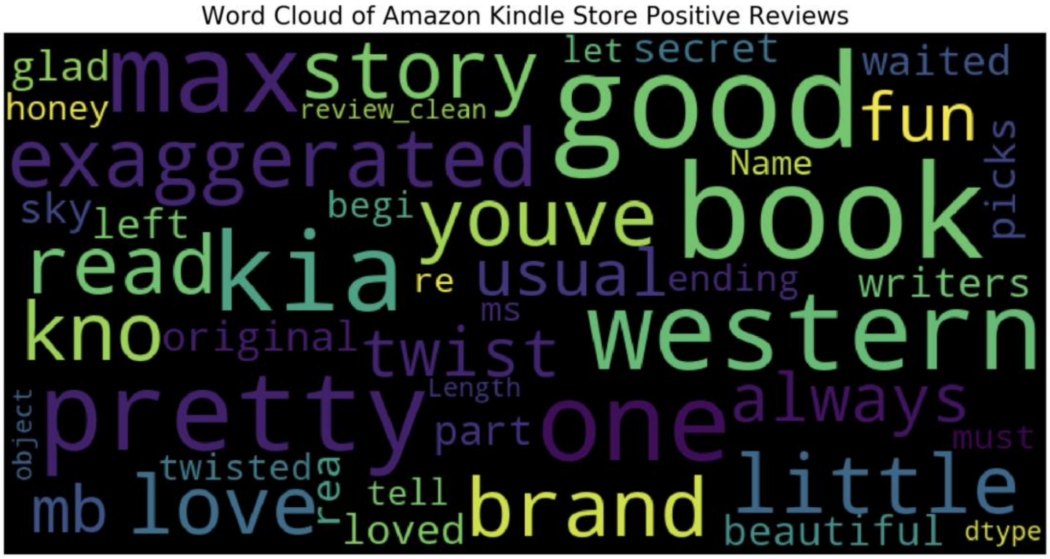


Fig 29: Positive Reviews

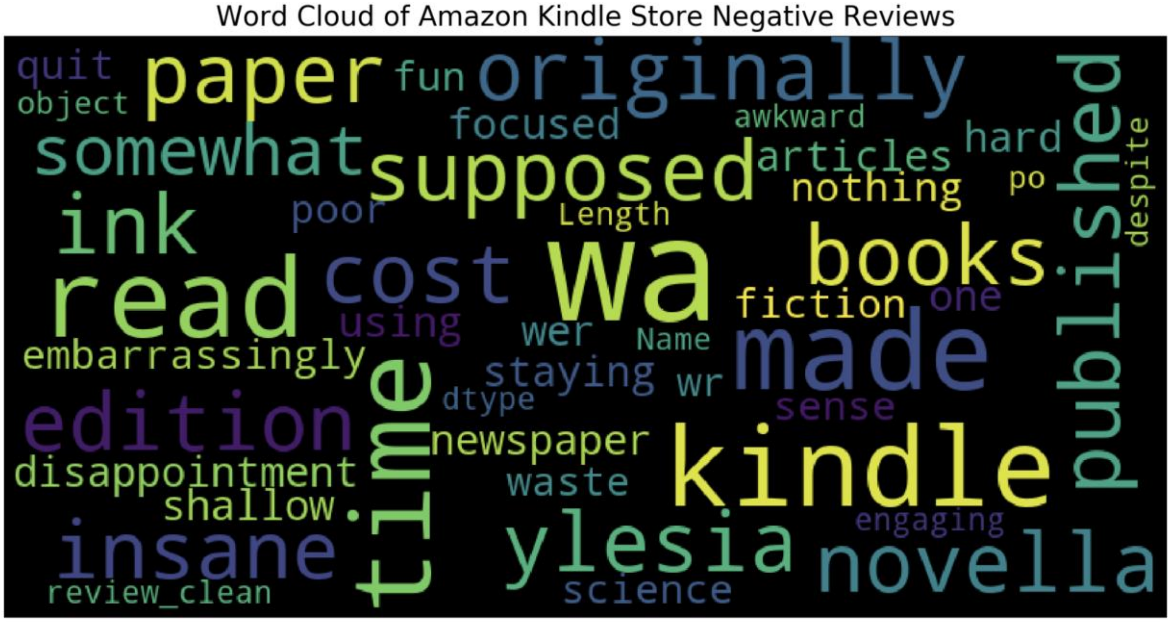


Fig 30: Negative Reviews

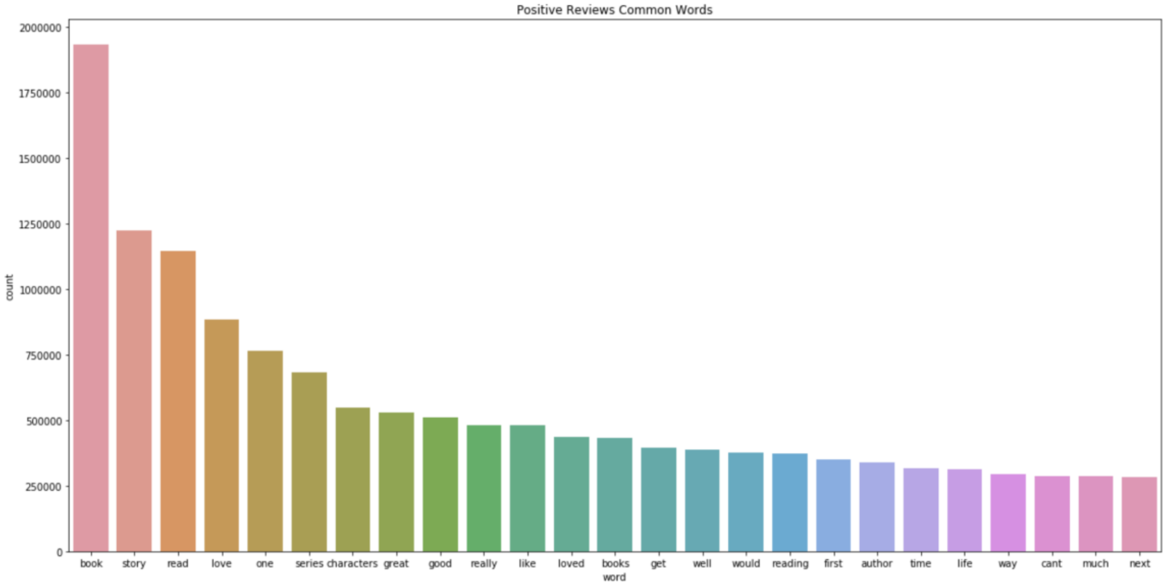


Fig 31: Positive common word

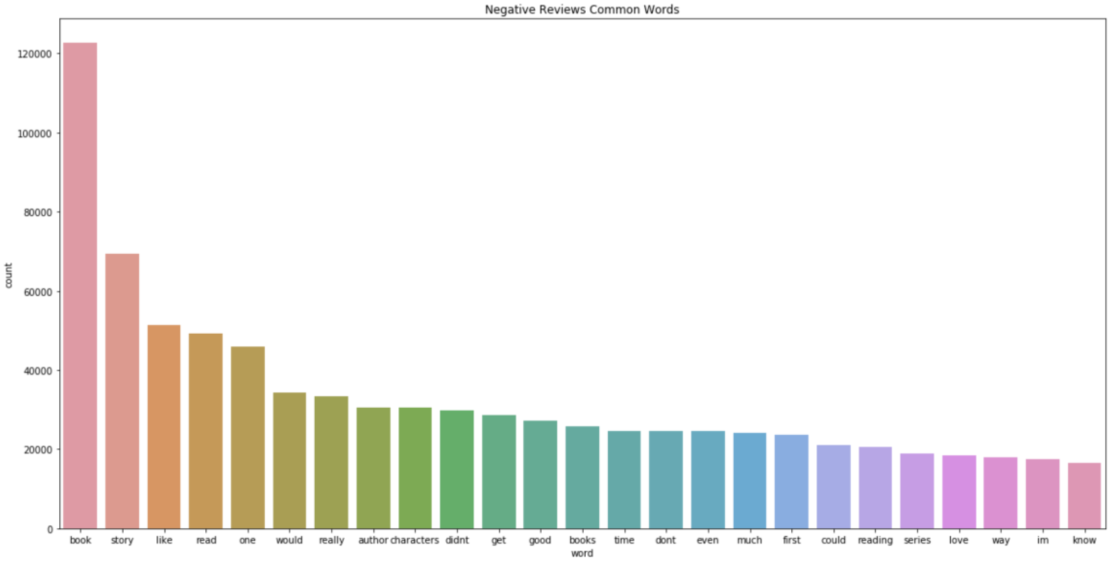


Fig 32: Top 25 common negative words

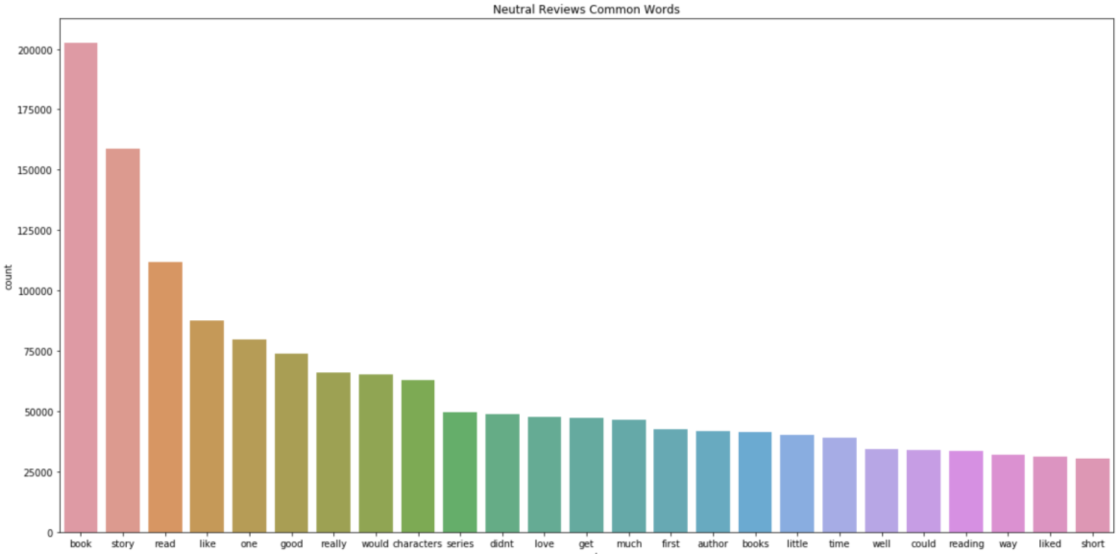
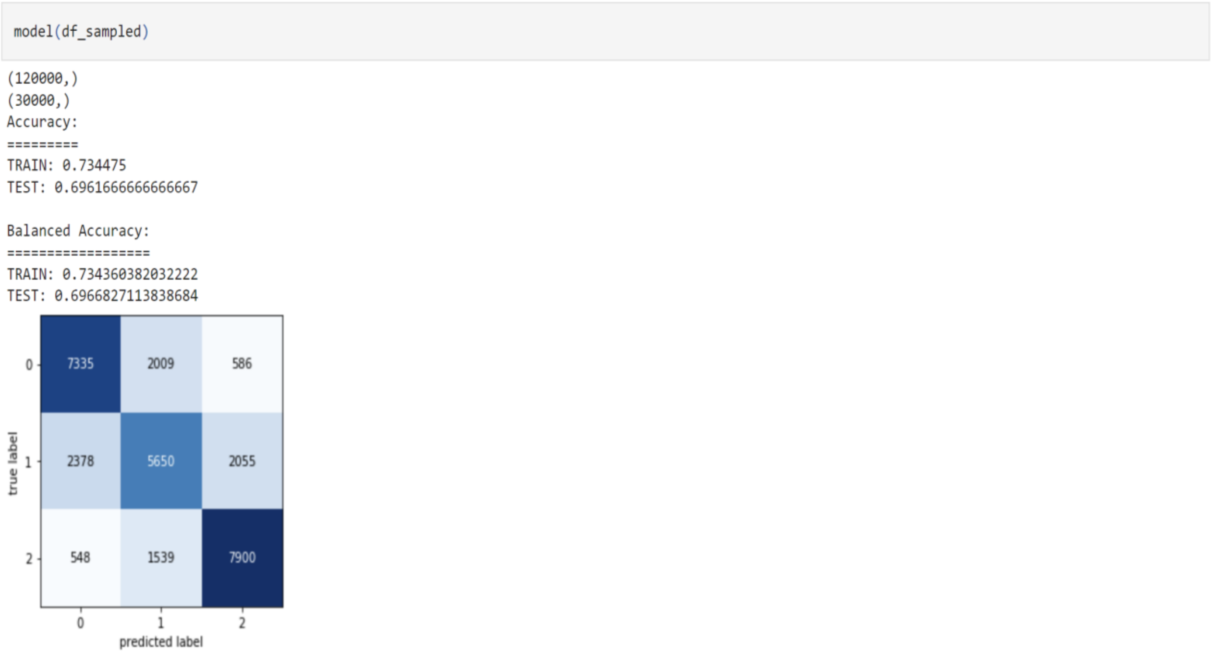


Fig 33: Top 25 common neutral words



```
model(df_sampled)
```

```
(120000,)
(30000,)
Accuracy:
=========
TRAIN: 0.734475
TEST: 0.6961666666666667

Balanced Accuracy:
==================
TRAIN: 0.734360382032222
TEST: 0.6966827113838684
```

This model is overfit and does not have the high precision required. We can adjust this model to handle this, but the main goal is to identify bad reviews, so we can also test deep learning models for this. We just tried LogReg to test a theory. We'll stop here and move on to the report's discussion of binary class machine learning and deep learning models.

3.3 CLEANING PROCESS

- There are numerous missing values in the data for certain features, and unnecessary columns exist for modeling purposes. The primary column we focus on is 'reviewText,' so we dropped the null values specifically for this column.
- Unwanted coloumns such as image, vote were removed.
- The reviews underwent a transformation where uppercase letters were converted to lowercase, and the punctuation marks were removed/cleaned.
- Rows that had a different style than the Kindle edition were removed.

3.4. PROPOSED MODEL USING 3CNN & 2RNN

The dataset provided consists of millions of reviews for Amazon products, which are categorized into positive and negative classes. These classes are evenly balanced in this dataset.

This dataset is substantial in size, and the version used for this project only includes the text itself without any additional metadata. This characteristic makes it an intriguing dataset for conducting natural language processing (NLP) tasks. Since the data is generated by users, it is expected to contain various typos, nonstandard spellings, and other variations that are not typically found in curated collections of published text.

A straightforward text processing approach will be applied to the dataset. Subsequently, two deep learning models, namely a convolutional neural network (CNN) and a recurrent neural network (RNN), will be implemented. It is important to note that these models are not heavily optimized. However, they are expected to yield results that are comparable, within a few percentage points, to state-of-the-art models in accurately predicting the binary sentiment of the reviews.

3.4.1. Reading the Text

Fortunately, the text is stored in a compressed format, allowing us to read it line by line. The first word of each line indicates the label, requiring us to convert it into a number. Subsequently, we consider the remaining portion as the comment or text content.

3.4.2. Text Preprocessing

To process the text, the initial step will involve converting all the text to lowercase and eliminating non-word characters. The non-word characters will be substituted with spaces, primarily since they mostly consist of punctuation marks. Additionally, any other characters, such as accented letters, will be removed. While it may be more appropriate to replace some of these characters with regular ASCII characters, that aspect will not be addressed in this context. It is worth noting that upon examining the character counts, it becomes apparent that this corpus contains minimal instances of unusual characters.

3.4.3. Splitting Data to Train and Test

In order to maintain consistency across all notebooks and ensure the utilization of the same sample data, a standardized approach is followed. The process involves consistently obtaining identical sample data and performing the data division using the same random state and test size for validation. By employing the same random state, it guarantees that the data splitting procedure yields identical training and validation subsets. This approach ensures that the experiments conducted in each notebook utilize the same data distribution, thereby reducing potential discrepancies and enabling fair comparisons and evaluations of different methodologies or models.

3.4.4. Train/Validation Split

We will allocate 20% of the training set for the purpose of validation.

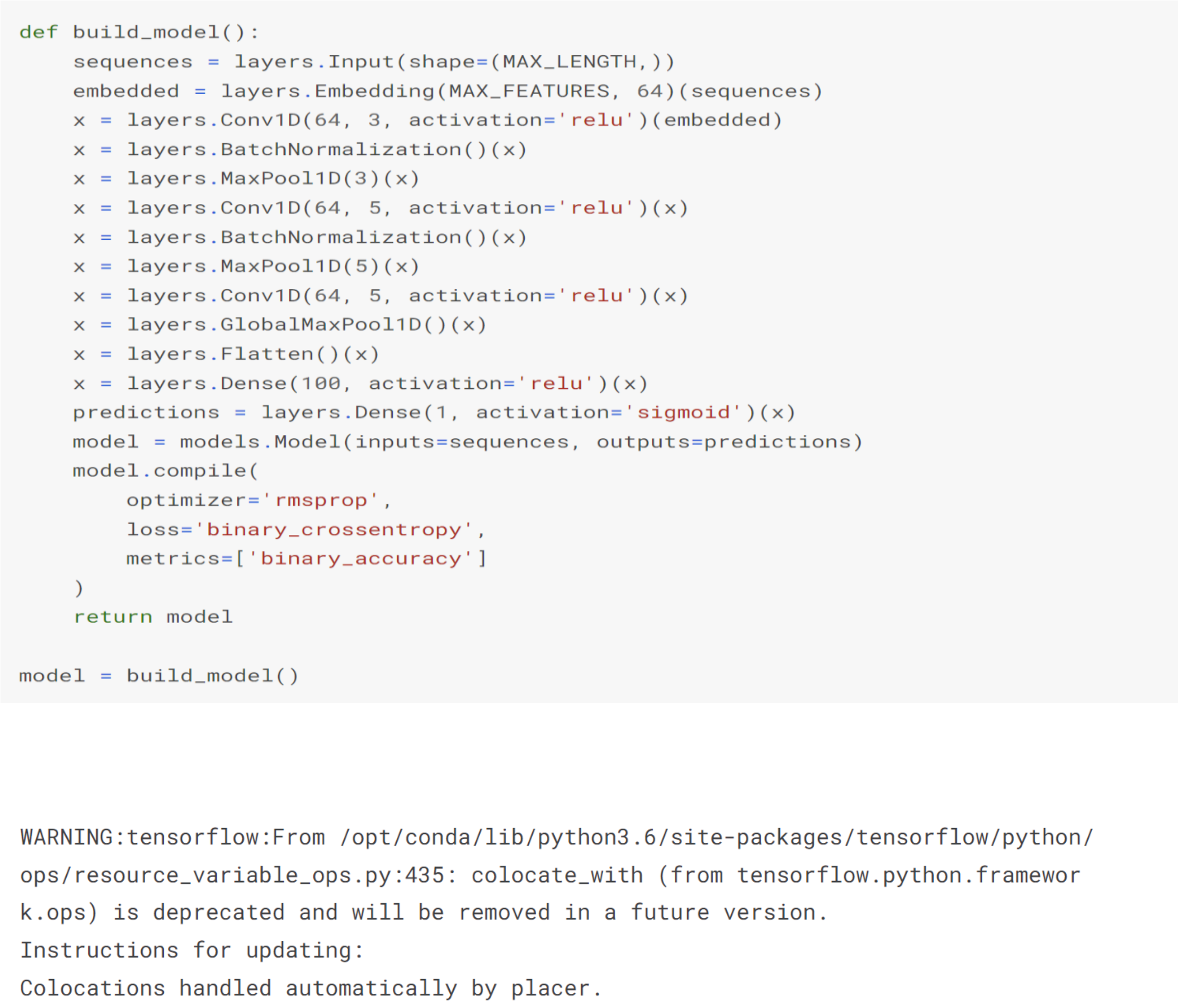
Keras offers tools that facilitate the conversion of text into formats suitable for deep learning models. Prior to this stage, some preprocessing has already been performed. Now, I will proceed by executing a Tokenizer, utilizing the top 12000 words as features.

3.4.5. Padding Sequences

To effectively utilize batches, we need to transform our sequences into sequences of equal length. Here, we will ensure that all sequences match the length of the longest sentence in the training set. Although we are not addressing it in this context, it can be beneficial to have variable sequence lengths, allowing each batch to contain sentences of similar lengths. This approach can help alleviate challenges associated with having excessive padded elements in a sequence. Additionally, there are various padding modes available that may prove useful for different models.

3.4.6. Convolutional Neural Net Model

We are employing relatively straightforward models for this task. In this particular CNN model, we utilize an embedding layer with a dimension of 64, followed by three convolutional layers. The first two convolutional layers incorporate batch normalization and max pooling, while the final convolutional layer uses global max pooling. The output of the convolutional layers is then passed to a dense layer, followed by the final output

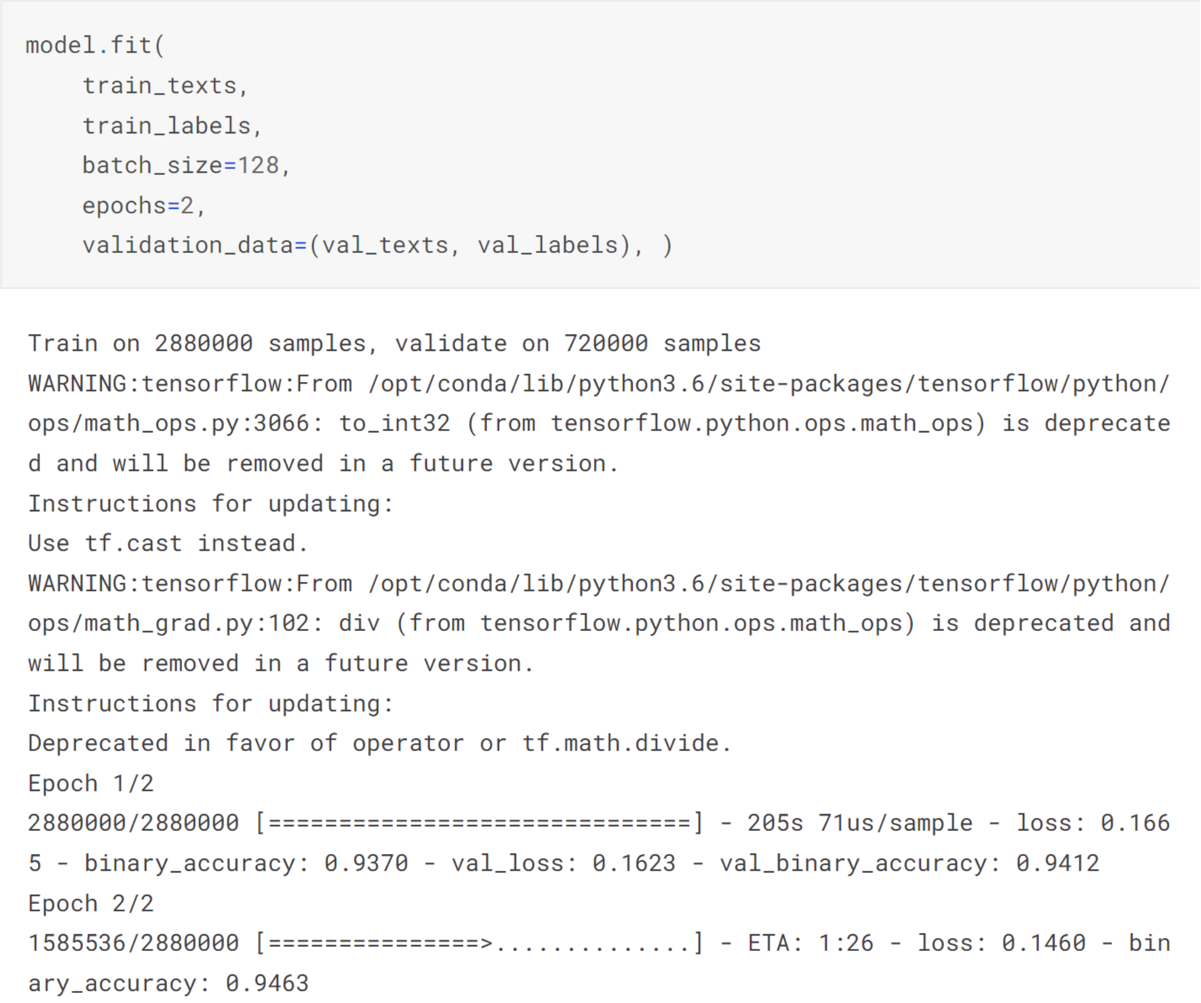


```
def build_model():
    sequences = layers.Input(shape=(MAX_LENGTH,))
    embedded = layers.Embedding(MAX_FEATURES, 64)(sequences)
    x = layers.Conv1D(64, 3, activation='relu')(embedded)
    x = layers.BatchNormalization()(x)
    x = layers.MaxPool1D(3)(x)
    x = layers.Conv1D(64, 5, activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.MaxPool1D(5)(x)
    x = layers.Conv1D(64, 5, activation='relu')(x)
    x = layers.GlobalMaxPool1D()(x)
    x = layers.Flatten()(x)
    x = layers.Dense(100, activation='relu')(x)
    predictions = layers.Dense(1, activation='sigmoid')(x)
    model = models.Model(inputs=sequences, outputs=predictions)
    model.compile(
        optimizer='rmsprop',
        loss='binary_crossentropy',
        metrics=['binary_accuracy']
    )
    return model

model = build_model()
```

```
WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/tensorflow/python/
ops/resource_variable_ops.py:435: colocate_with (from tensorflow.python.framewor
k.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
```

Fig 34: CNN model



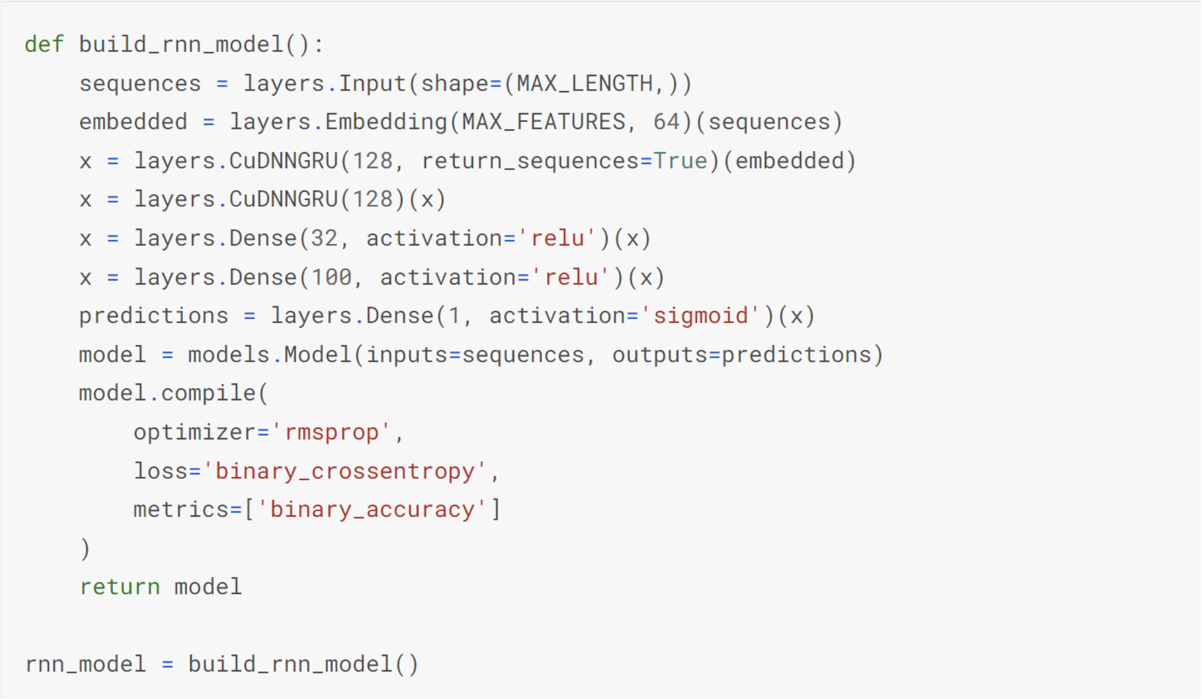
```
model.fit(
    train_texts,
    train_labels,
    batch_size=128,
    epochs=2,
    validation_data=(val_texts, val_labels), )
```

```
Train on 2880000 samples, validate on 720000 samples
WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/tensorflow/python/
ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecate
d and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/tensorflow/python/
ops/math_grad.py:102: div (from tensorflow.python.ops.math_ops) is deprecated and
will be removed in a future version.
Instructions for updating:
Deprecated in favor of operator or tf.math.divide.
Epoch 1/2
2880000/2880000 [==============================] - 205s 71us/sample - loss: 0.166
5 - binary_accuracy: 0.9370 - val_loss: 0.1623 - val_binary_accuracy: 0.9412
Epoch 2/2
1585536/2880000 [===============>..............] - ETA: 1:26 - loss: 0.1460 - bin
ary_accuracy: 0.9463
```

Fig 35: Fitting data to my model

3.4.7. Recurrent Neural Net Model

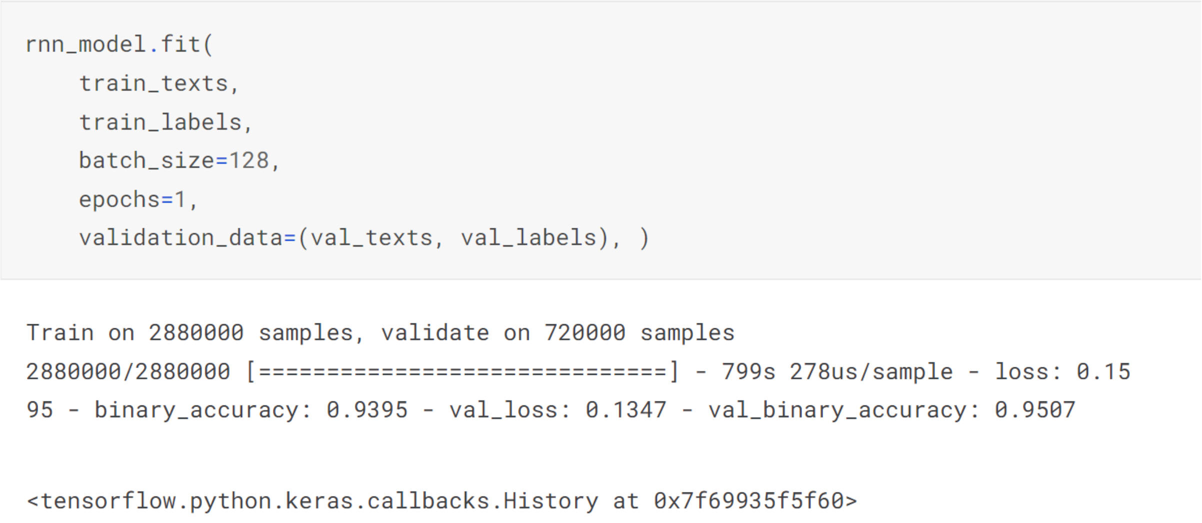
For the RNN model, we are opting for a simple architecture. It consists of an embedding layer, two layers of GRU, followed by two dense layers, and ultimately the output layer. To optimize performance, we employ the CuDNNGRU instead of the regular GRU, as it offers significantly faster computation, potentially over ten times faster, particularly on Kaggle's servers.



```
def build_rnn_model():
    sequences = layers.Input(shape=(MAX_LENGTH,))
    embedded = layers.Embedding(MAX_FEATURES, 64)(sequences)
    x = layers.CuDNNGRU(128, return_sequences=True)(embedded)
    x = layers.CuDNNGRU(128)(x)
    x = layers.Dense(32, activation='relu')(x)
    x = layers.Dense(100, activation='relu')(x)
    predictions = layers.Dense(1, activation='sigmoid')(x)
    model = models.Model(inputs=sequences, outputs=predictions)
    model.compile(
        optimizer='rmsprop',
        loss='binary_crossentropy',
        metrics=['binary_accuracy']
    )
    return model

rnn_model = build_rnn_model()
```

Fig 36: RNN model



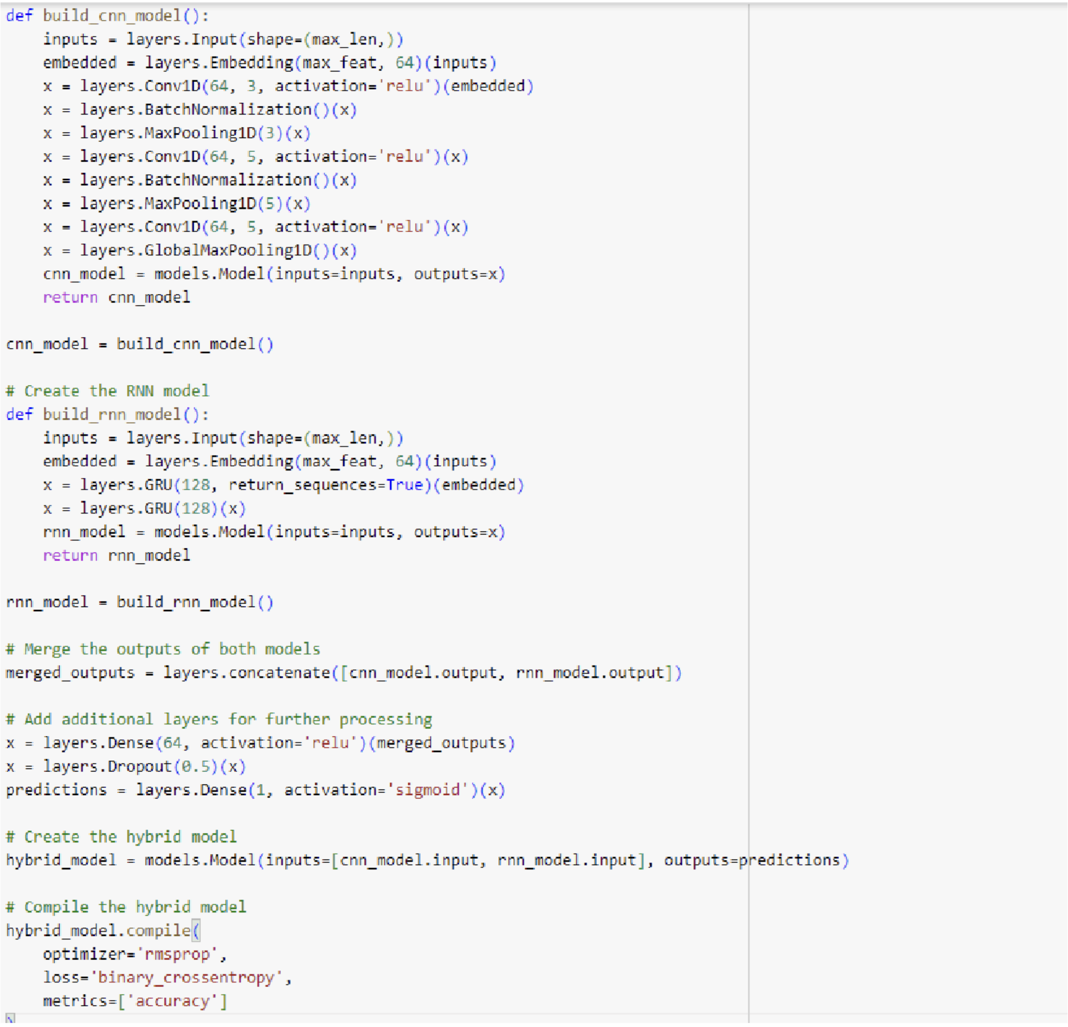
```
rnn_model.fit(
    train_texts,
    train_labels,
    batch_size=128,
    epochs=1,
    validation_data=(val_texts, val_labels), )
```

```
Train on 2880000 samples, validate on 720000 samples
2880000/2880000 [==============================] - 799s 278us/sample - loss: 0.15
95 - binary_accuracy: 0.9395 - val_loss: 0.1347 - val_binary_accuracy: 0.9507


<tensorflow.python.keras.callbacks.History at 0x7f69935f5f60>
```

Fig 37: Fitting data to my model

3.4.8. Hybrid 3CNN 2RNN Model



```
def build_cnn_model():
    inputs = layers.Input(shape=(max_len,))
    embedded = layers.Embedding(max_feat, 64)(inputs)
    x = layers.Conv1D(64, 3, activation='relu')(embedded)
    x = layers.BatchNormalization()(x)
    x = layers.MaxPooling1D(3)(x)
    x = layers.Conv1D(64, 5, activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.MaxPooling1D(5)(x)
    x = layers.Conv1D(64, 5, activation='relu')(x)
    x = layers.GlobalMaxPooling1D()(x)
    cnn_model = models.Model(inputs=inputs, outputs=x)
    return cnn_model

cnn_model = build_cnn_model()

# Create the RNN model
def build_rnn_model():
    inputs = layers.Input(shape=(max_len,))
    embedded = layers.Embedding(max_feat, 64)(inputs)
    x = layers.GRU(128, return_sequences=True)(embedded)
    x = layers.GRU(128)(x)
    rnn_model = models.Model(inputs=inputs, outputs=x)
    return rnn_model

rnn_model = build_rnn_model()

# Merge the outputs of both models
merged_outputs = layers.concatenate([cnn_model.output, rnn_model.output])

# Add additional layers for further processing
x = layers.Dense(64, activation='relu')(merged_outputs)
x = layers.Dropout(0.5)(x)
predictions = layers.Dense(1, activation='sigmoid')(x)

# Create the hybrid model
hybrid_model = models.Model(inputs=[cnn_model.input, rnn_model.input], outputs=predictions)

# Compile the hybrid model
hybrid_model.compile(
    optimizer='rmsprop',
    loss='binary_crossentropy',
    metrics=['accuracy']
```

Fig 38: Hybrid Model

3.5. PROPOSED MODEL USING GNN

A message evaluation structure in light of graph neural networks and long short-term memory (GNN-LSTM) is created and tried in this review to resolve this issue. Syntactic analysis really holds the semantic standards and primary linkages of short texts, and an unstructured component extraction is finished utilizing a GNN semantic parser. The combination method does feel examination utilizing a syntax tree and a GNN, which has hypothetical and commonsense ramifications for working on the exhibition of short text sentiment classification tasks.

Advantages:

The model has good generalizability for the Internet's open comment area and can conduct sentiment analysis on short messages.

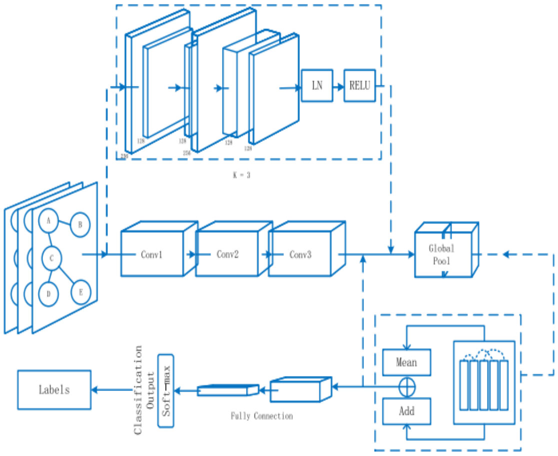


Fig.2: System architecture

MODULES:

We designed the modules indicated below to carry out the aforementioned project.

- Data exploration: we will use this module to enter data into the system.
- This module will be used to read data for processing.
- This module will divide the data into train and test segments.
- Building the model - - GCN - GCN with BERT - GRU - LSTM - CNN - Bi- LSTM - BERT GCN + LSTM + CNN - BERT GCN + LSTM.
- User registration and login: Using this module requires registration and login.
- The use of this module will result in anticipated input.
- Prediction: the final predicted value is shown.

3.5. SVM BASED MODEL

This work presents a creamer game plan that joins the Support Vector Machine (SVM) calculation with Particle Swarm Optimization (PSO) and other oversampling techniques to resolve the issue of imbalanced data. By refreshing the dataset, which incorporates overviews from various Jordanian restaurants, SVM is utilized as a ML gathering system to manage review feelings. The data was gotten from Jeeran, a notable casual Arabic test bunch. Four unique oversampling methodologies, explicitly the Synthetic Minority Oversampling Technique (SMOTE), SVM-SMOTE, Adaptive Synthetic Sampling (ADASYN), and borderline-SMOTE are used to give an improved dataset and address the dataset's imbalanced issue. A PSO strategy is utilized to work on the heaps of the components.

Advantages:

When compared to other classification algorithms, the suggested PSO-SVM strategy delivers the best results.

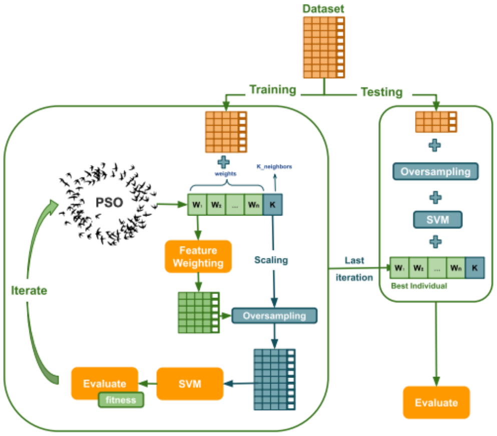


Fig.2: System architecture

MODULES:

To carry out the aforementioned project, we created the modules listed below.

- Information investigation: we will place information into the framework utilizing this module.

- Handling: we will peruse information for handling utilizing this module.

- Utilizing this module, information will be isolated into train and test.

- Model generation: Building the model - Bi-LSTM - Ri-RNN - Bi-GRU - GRU - RNN - LSTM - CNN - LSTM + GRU with CNN. Calculated algorithm accuracy

- User signup and login: Using this module will result in registration and login.

- User input: Using this module will result in predicted input.

- Prediction: the final predicted value is shown.

CHAPTER 4

RESULTS

5.1 MODEL EVALUATION

We use the following measurements based on confusion matrices conclusions for predictionevaluation.

5.1.1 Precision

Simply expressed, precision refers to the percentage of actual positive results out of the totalpositive anticipated by the model.
The following equation can be used to calculate precision values:

5.1.2 Recall

The rate at which the system is able to relearn information is referred to as recall. As an outcome, Recall estimates how many true positive (TP) traits our model identified and labeledas positive.

The following equation can be used to calculate recall values:

$$\text{Recall} = \frac{Sum\ x\ in\ \mathrm{K}\ TruePositives\_x}{Sum\ x\ in\ \mathrm{K}\ (TruePositives\_x + FalseNegatives\_x)}$$

5.1.3 F1-score

A significant number of True Negatives (TN), which in most business situations do not rely onmuch, contribute to the accuracy, although False Negatives (FN) and False Positives

(FP) frequently have business consequences. If we need to find the right balance between Recalland Precision There is an unequal class distribution, F1-Score would be an appropriate statistic to utilize.

$$\text{F1-Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5.1.4 Accuracy

The data that is correctly categorized divided by the whole dataset evaluated is how accuracy iscalculated. It can also be calculated as a 1-error.
The following equation can be used to calculate the accuracy:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

5.1.5 Model prediction

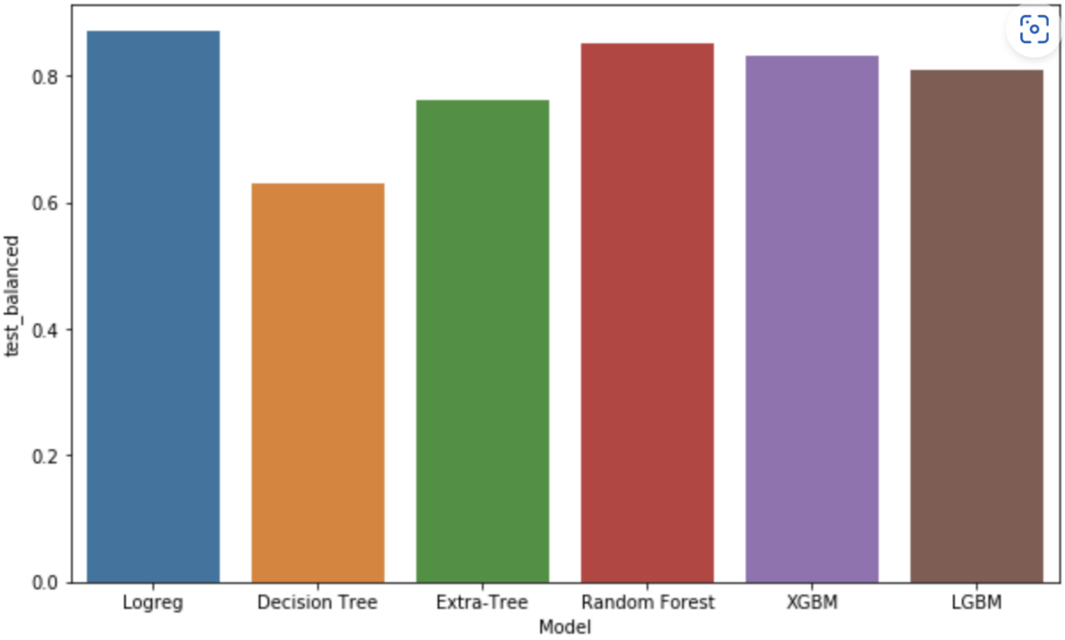
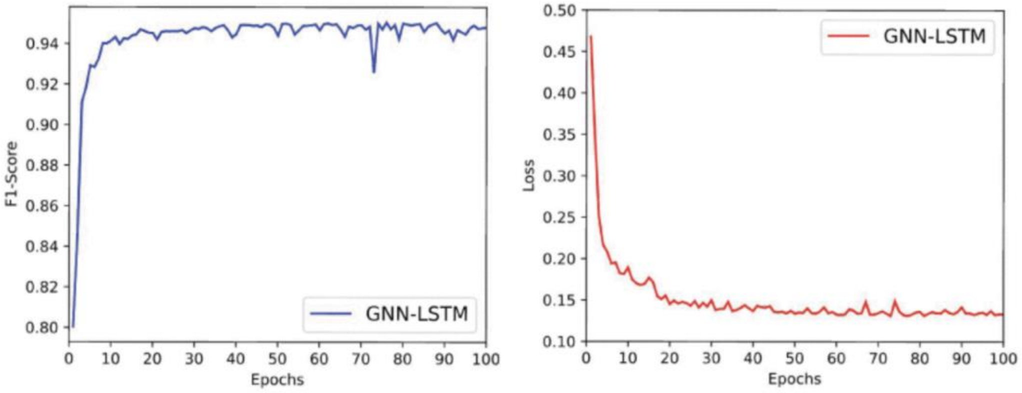


Fig 39: accuracy achieved in ML models

We can interpret from the graph that “Logistic Regression” performs the best with test accuracy of 0.870 followed by random forest and XGBM while decision tree performs the worst with test accuracy of 0.628 for the dataset.

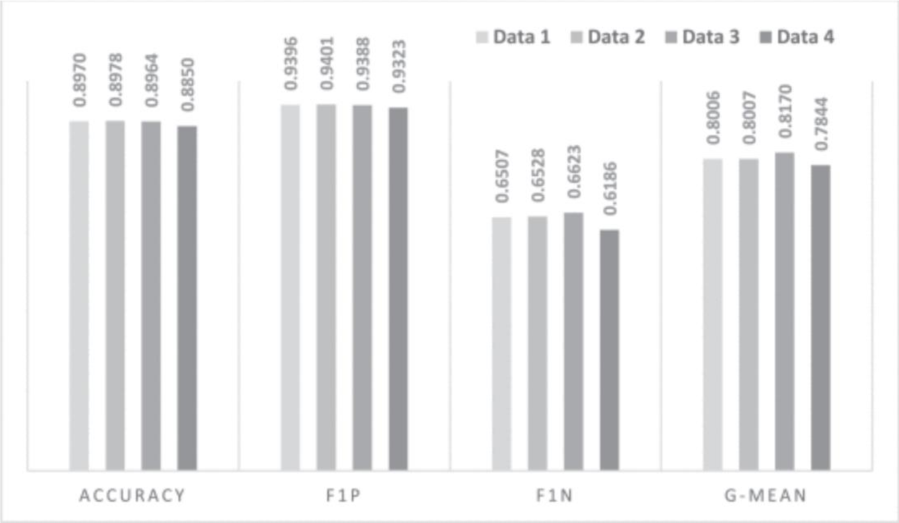
More work on fine-tuning can be done in these models to produce better outcomes. But we would like to try and see how deep learning models will do.

Results for GNN based approach: -



GNN-LSTM’s performance in the Weibo comments dataset, according to the experimental results, with an F1 score of 95.22 percent and an accuracy of 95.25 percent.

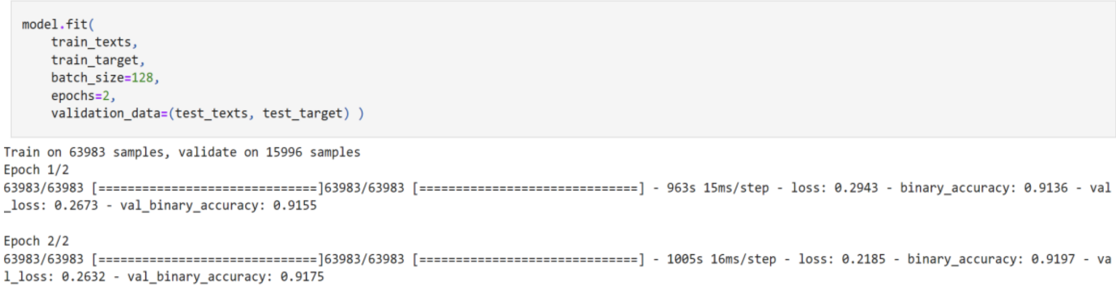
Results for SVM based approach:-



Comparative results of different datasets for the proposed PSO-SVM model.

Here we can clearly see that the maximum accuracy achieved using this model is 93.96%

CNN model :-



```
model.fit(
    train_texts,
    train_target,
    batch_size=128,
    epochs=2,
    validation_data=(test_texts, test_target) )
```

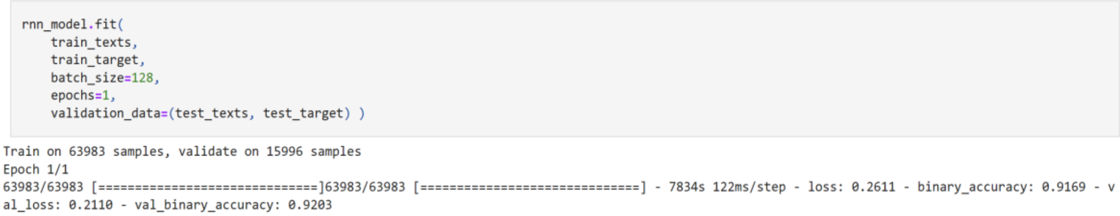
```
Train on 63983 samples, validate on 15996 samples
Epoch 1/2
63983/63983 [==============================]63983/63983 [==============================] - 963s 15ms/step - loss: 0.2943 - binary_accuracy: 0.9136 - val
_loss: 0.2673 - val_binary_accuracy: 0.9155

Epoch 2/2
63983/63983 [==============================]63983/63983 [==============================] - 1005s 16ms/step - loss: 0.2185 - binary_accuracy: 0.9197 - va
l_loss: 0.2632 - val_binary_accuracy: 0.9175
```

Fig 40: Accuracy achieved using CNN model

Model provides 91.97% accuracy on validation data.

RNN model :-



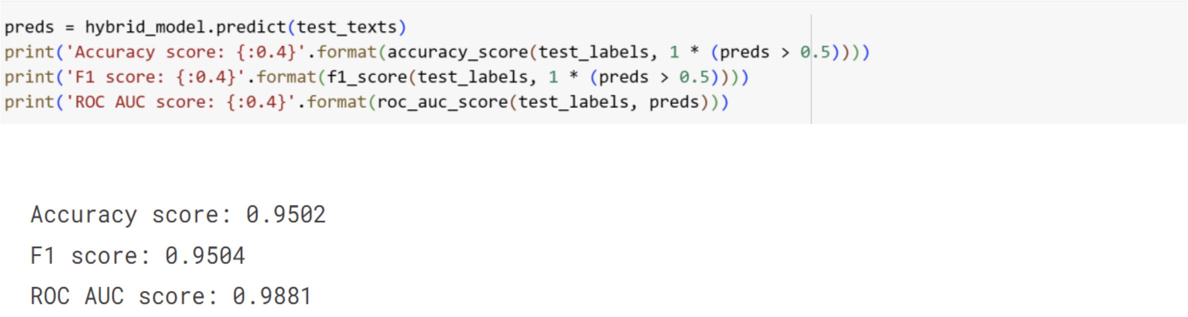
```
rnn_model.fit(
    train_texts,
    train_target,
    batch_size=128,
    epochs=1,
    validation_data=(test_texts, test_target) )
```

```
Train on 63983 samples, validate on 15996 samples
Epoch 1/1
63983/63983 [==============================]63983/63983 [==============================] - 7834s 122ms/step - loss: 0.2611 - binary_accuracy: 0.9169 - v
al_loss: 0.2110 - val_binary_accuracy: 0.9203
```

Fig 41: Accuracy achieved using RNN model

Model provides 92.03% accuracy on validation data.

3CNN + 2RNN Hybrid model :-



```
preds = hybrid_model.predict(test_texts)
print('Accuracy score: {:0.4}'.format(accuracy_score(test_labels, 1 * (preds > 0.5))))
print('F1 score: {:0.4}'.format(f1_score(test_labels, 1 * (preds > 0.5))))
print('ROC AUC score: {:0.4}'.format(roc_auc_score(test_labels, preds)))
```

```
Accuracy score: 0.9502
F1 score: 0.9504
ROC AUC score: 0.9881
```

Fig 42: accuracy achieved using RNN model

CHAPTER 5

CONCLUSION

This work mainly focused on the overall idea of sentiment analysis such as understanding different types of analysis that can be performed on data present online, various methods of pre-processing data to increase the performance of classifier, getting overview of machine learning and its algorithm, developing the basic understanding definitions and words used commonly during analysis like POS tagger, Bag-of-words and more. Most challenging part of sentiment analysis is dealing with text having irony and sarcasm or statement where comparison is done between products.

As far as exactness, F-measure and AUC, the review exhibits that the proposed PSO-SVM technique beats choices in all test boundaries. The PSO-SVM beat the standard SVM, LR, RF, DT, k-NN, and XGBoost in each dataset variety. In the future, we intend to utilize a variety of metaheuristic algorithms with this data. It is also possible to forecast how reviews will be accepted using other applications, such as those in the medical and technical fields.

GNN-LSTM finds and improves persistent elements for sentiment analysis. The LSTM model channels out chart commotion while holding helpful totaled data. The exploratory findings demonstrate that the GNN-LSTM model can lead evaluation with brief comments and is extremely generalizable to the open comment section of the Internet.

This project has focused on the application of a hybrid 3CNN-2RNN model in sentiment analysis, which has proven to be highly effective. The model combines the strengths of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to achieve improved results compared to using CNN or RNN alone.

By leveraging the capabilities of both CNNs and RNNs, the hybrid model takes advantage of their respective features. CNNs excel at capturing local patterns and extracting meaningful features from text data, while RNNs are proficient in modeling sequential dependencies and capturing long-term dependencies in the data.

The experimental results of this project demonstrate that the hybrid 3CNN-2RNN model outperforms both the CNN and RNN models individually, yielding a significant 2% improvement in accuracy. This improvement showcases the effectiveness of leveraging the combined power of CNNs and RNNs in sentiment analysis tasks. Deep learning models performed better than supervised machine learning models while analyzing the reviews.

Future Improvements

We can try the suggestions below to improve the performance of the model;

1. Models for gradient boosts can be adjusted.
2. Grid search is possible.
3. When we return to the text column, we may perform more thorough analysis, cleaning, and feature engineering.

There are numerous unexplored possibilities in this context. We believe that the original data from Amazon contains additional fields that could be incorporated into the model. Furthermore, we have yet to include global features derived from the samples, such as length and character-level features, among others. It is even plausible to explore the implementation of deep learning models at the character level, potentially reducing sensitivity to misspellings. In the case of online reviews, character-level features hold considerable significance as users may intentionally misspell words to evade moderation. However, considering that the current models are already achieving accuracy well above 90%, any further improvements from this point onwards are likely to be marginal.

REFERENCES

[1]. Chan, Jireh Yi-Le, Khean Thye Bea, Steven Mun Hong Leow, Seuk Wai Phoong, and Wai Khuen Cheng. "State of the art: a review of sentiment analysis based on sequential transfer learning." *Artificial Intelligence Review* 56, no. 1 (2023): 749-780.

[2]. Suhendra, Nikmatul Husna Binti, Pantea Keikhosrokiani, Moussa Pourya Asl, and Xian Zhao. "Opinion mining and text analytics of literary reader responses: A case study of reader responses to KL Noir volumes in Goodreads using sentiment analysis and topic." In *Handbook of Research on Opinion Mining and Text Analytics on Literary Works and Social Media*, pp. 191-239. IGI Global, 2022.

[3]. Mridula, A., and C. R. Kavitha. "Opinion mining and sentiment study of tweets polarity using machine learning." In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 621-626. IEEE, 2018.

[4]. Ramachandran, Dharini, and R. Parvathi. "Analysis of twitter specific preprocessing technique for tweets." *Procedia Computer Science* 165 (2019): 245-251.

[5]. R. Obiedat *et al*., "Sentiment Analysis of Customers' Reviews Using a Hybrid Evolutionary SVM- Based Approach in an Imbalanced Data Distribution," in *IEEE Access*, vol. 10, pp. 22260- 22273, 2022

[6]. S. Dhar, S. Pednekar, K. Borad, and A. Save, "Sentiment Analysis Using Neural Networks: A New Approach,"

[7]. Duncan, Brett, and Yanqing Zhang. "Neural networks for sentiment analysis on Twitter." In *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pp. 275-278. IEEE, 2015.

[8]. Akilandeswari, J., and G. Jothi. "Sentiment classification of tweets with non-language features." *Procedia Computer Science* 143 (2018): 426-433.

[9]. Gokulakrishnan, Balakrishnan, Pavalanathan Priyanthan, Thiruchittampalam Ragavan, Nadarajah Prasath, and AShehan Perera. "Opinion mining and sentiment analysis on a twitter data stream." In *International conference on advances in ICT for emerging regions (ICTer2012)*, pp. 182-188. IEEE, 2012.

[10]. Ahuja, Ravinder, Aakarsha Chug, Shruti Kohli, Shaurya Gupta, and Pratyush Ahuja. "The impact of features extraction on the sentiment analysis." *Procedia Computer Science* 152 (2019): 341-348.

[11]. Hatzivassiloglou, Vasileios, and Kathleen McKeown. "Predicting the semantic orientation of adjectives." In *35th annual meeting of the association for computational linguistics and 8th conference of the european chapter of the association for computational linguistics*, pp. 174-181. 1997.

[12]. Fang, Xing, and Justin Zhan. "Sentiment analysis using product review data." *Journal of Big Data* 2, no. 1 (2015): 1-14.

[13]. Haddi, Emma, Xiaohui Liu, and Yong Shi. "The role of text pre-processing in sentiment analysis." *Procedia computer science* 17 (2013): 26-32.

[14]. Y. Li and N. Li, "Sentiment Analysis of Weibo Comments Based on Graph Neural Network," in IEEE Access, vol. 10, pp. 23497-23510, 2022.

[15]. Agarwal, Apoorv, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca J. Passonneau. "Sentiment analysis of twitter data." In *Proceedings of the workshop on language in social media (LSM 2011)*, pp. 30-38. 2011.

[16]. Saif, Hassan, Yulan He, and Harith Alani. "Semantic sentiment analysis of twitter." In *International semantic web conference*, pp. 508-524. Springer, Berlin, Heidelberg, 2012.

[17]. SEBASTIAN, TEEJA. "Sentiment Analysis for Twitter." PhD diss., 2012.

[18]. Kouloumpis, Efthymios, Theresa Wilson, and Johanna Moore. "Twitter sentiment analysis: The good the bad and the omg!." In *Proceedings of the international AAAI conference on web and social media*, vol. 5, no. 1, pp. 538-541. 2011.

[19]. Neethu, M. S., and R. Rajasree. "Sentiment analysis in twitter using machine learning techniques." In *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*, pp. 1-5. IEEE, 2013.

[20]. Birjali, Marouane, Mohammed Kasri, and Abderrahim Beni-Hssane. "A comprehensive survey on sentiment analysis: Approaches, challenges and trends." *Knowledge-Based Systems* 226 (2021): 107134.

[21]. Yadav, Ashima, and Dinesh Kumar Vishwakarma. "Sentiment analysis using deep learning architectures: a review." *Artificial Intelligence Review* 53, no. 6 (2020): 4335- 4385

[22]. Phan, Huyen Trang, Van Cuong Tran, Ngoc Thanh Nguyen, and Dosam Hwang. "Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model." *IEEE Access* 8 (2020): 14630-14641.

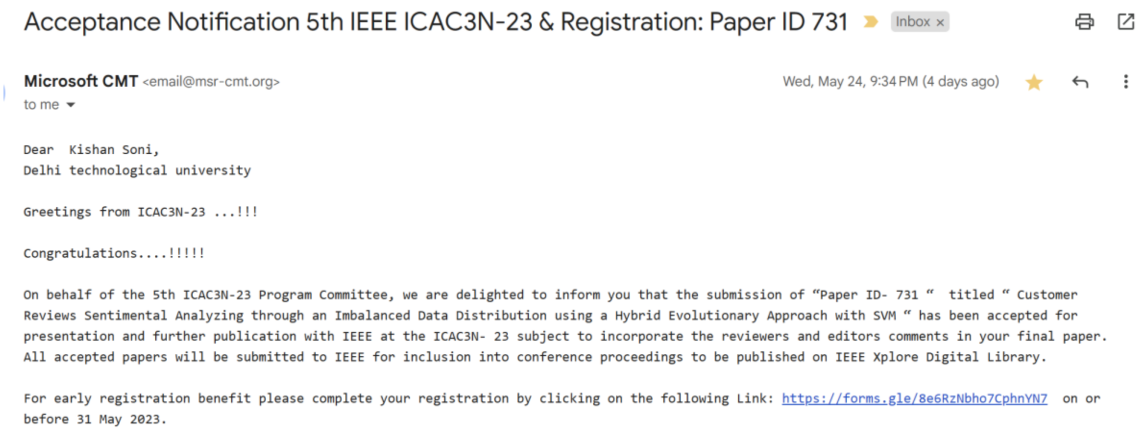
[23]. Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." *Ain Shams engineering journal* 5, no. 4 (2014): 1093-1113.

[24]. Zhang, Lei, Shuai Wang, and Bing Liu. "Deep learning for sentiment analysis: A survey." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, no. 4 (2018): e1253.

[25]. Al-Natour, Sameh, and Ozgur Turetken. "A comparative assessment of sentiment analysis and star ratings for consumer reviews." *International Journal of Information Management* 54 (2020): 102132.

LIST OF PUBLICATION

1. Kishan Soni and Manoj Sethi “Customer Reviews Sentimental Analyzing through an Imbalanced Data Distribution using a Hybrid Evolutionary Approach with SVM”, 2023 5th International Conference on Advances In Computing, Communication Control and Networking (ICAC3N)



Acceptance Notification 5th IEEE ICAC3N-23 & Registration: Paper ID 731

Microsoft CMT <email@msr-cmt.org>
to me

Wed, May 24, 9:34 PM (4 days ago)

Dear Kishan Soni,
Delhi technological university

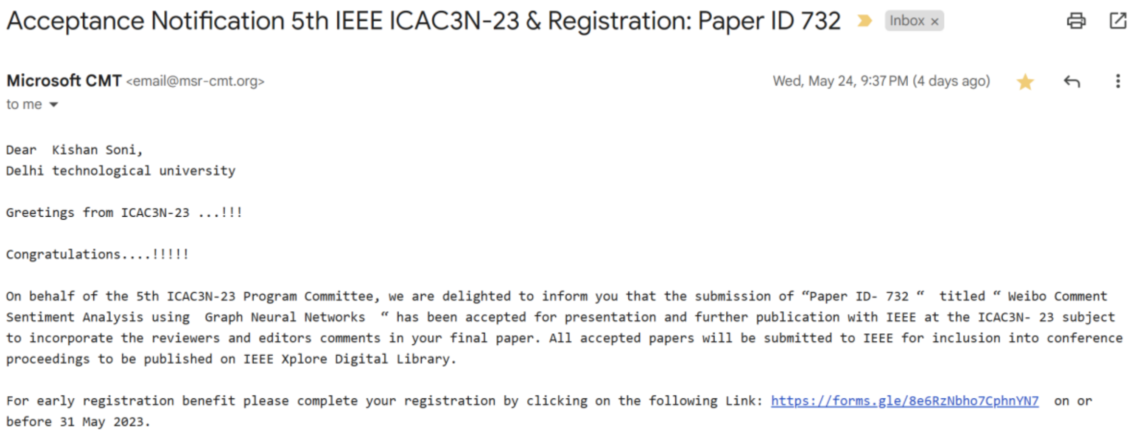
Greetings from ICAC3N-23 ...!!!

Congratulations....!!!!!

On behalf of the 5th ICAC3N-23 Program Committee, we are delighted to inform you that the submission of “Paper ID- 731 “ titled “ Customer Reviews Sentimental Analyzing through an Imbalanced Data Distribution using a Hybrid Evolutionary Approach with SVM “ has been accepted for presentation and further publication with IEEE at the ICAC3N- 23 subject to incorporate the reviewers and editors comments in your final paper. All accepted papers will be submitted to IEEE for inclusion into conference proceedings to be published on IEEE Xplore Digital Library.

For early registration benefit please complete your registration by clicking on the following Link: https://forms.gle/8e6RzNbho7CphnYN7 on or before 31 May 2023.

2. Kishan Soni and Manoj Sethi “Weibo Comment Sentiment Analysis using Graph Neural Networks”, 2023 5th International Conference on Advances In Computing, Communication Control and Networking (ICAC3N)



Acceptance Notification 5th IEEE ICAC3N-23 & Registration: Paper ID 732

Microsoft CMT <email@msr-cmt.org>
to me

Wed, May 24, 9:37 PM (4 days ago)

Dear Kishan Soni,
Delhi technological university

Greetings from ICAC3N-23 ...!!!

Congratulations....!!!!!

On behalf of the 5th ICAC3N-23 Program Committee, we are delighted to inform you that the submission of “Paper ID- 732 “ titled “ Weibo Comment Sentiment Analysis using Graph Neural Networks “ has been accepted for presentation and further publication with IEEE at the ICAC3N- 23 subject to incorporate the reviewers and editors comments in your final paper. All accepted papers will be submitted to IEEE for inclusion into conference proceedings to be published on IEEE Xplore Digital Library.

For early registration benefit please complete your registration by clicking on the following Link: https://forms.gle/8e6RzNbho7CphnYN7 on or before 31 May 2023.