Sentiment Analysis on Drug Reviews using BERT with CNN Model.

Sarthak Bhatnagar
MSc. in Data Analytics
(School of Computing)
National College of Ireland
Dublin, Ireland
x21185352@student.ncirl.ie

Utkarsh Singh
MSc. in Data Analytics
(School of Computing)
National College of Ireland
Dublin, Ireland
x21199922@student.ncirl.ie

Ronit Kumar Pareta
MSc. in Data Analytics
(School of Computing)
National College of Ireland
Dublin, Ireland
x21223645@student.ncirl.ie

Rohit Roy
MSc. in Data Analytics
(School of Computing)
National College of Ireland
Dublin, Ireland
x21201102@student.ncirl.ie

Abstract— **In the surveillance of drugs and drug development, sentiment analysis of drug evaluations using natural language processing (NLP) is a crucial task. In this paper, we use NLP approaches to propose a thorough method for sentiment analysis of medication reviews. This report presents the findings of a project that aimed to analyze drug reviews provided by patients on various health forums. The project's primary objective was to extract useful information from the data and visualize patterns in the data to gain insights into patients' preferences, demographics, drug effectiveness, and associated side effects. Our findings demonstrate that our method achieves excellent classification accuracy for sentiment, and it can be a helpful tool for pharmaceutical firms, medical experts, and patients to measure the general sentiment of medications.**
Keywords—CNN, BERT Model, NLP, Tensorflow and Sentiment Analysis.

I. INTRODUCTION

Drug review analysis is a critical process that evaluates the safety, efficacy, and other factors of a drug. The objective of drug review analysis is to gather and evaluate all available data about drugs and determine whether they are safe and effective for their intended use. In this report, we will review a dataset that comprises multiple drugs and discuss why it is important to analyse such datasets and customer sentiments over the drugs usage. The dataset we have chosen contains reviews of multiple drugs that are commonly used to treat various medical conditions. The reviews are from real patients who have used the drugs, and they provide valuable information about the drugs' effectiveness and safety. By analysing this dataset, we can gain insights into the drugs' performance and identify any common issues or concerns that patients have with them.

Despite the widespread use of these drugs, there may be limited information available on their real-world performance and side effects. By analysing this dataset, we can fill this knowledge gap and provide healthcare professionals and patients with valuable information that can help them make informed decisions about the use of these drugs. In addition to our standard drug review analysis, we will also apply sentiment analysis to the dataset. Sentiment analysis is a process that uses natural language processing techniques to determine the emotional tone of a text. By applying sentiment analysis to the patient reviews in the dataset, we can identify any common themes or issues that patients have with the drugs and gain insights into their overall satisfaction with the drugs. This information can be used to improve the drugs' performance and patient outcomes. In this report, we will provide a comprehensive analysis of the dataset, including the drugs' background, their mechanism of action, indications, adverse effects, contraindications, drug interactions, and other factors that are relevant to their safe and effective use. We will also present the results of our sentiment analysis and discuss their implications for healthcare professionals and patients.

This project is about analysing drug reviews provided by patients on various health forums. The purpose of this report is to present the findings of the project and visualize the patterns in the data to gain insights about patients' preferences, demographics, the effectiveness of the drugs, and the side effects associated with them. The dataset used in the project consists of two parts: Drug_Train_Excel.xlsx and Drug_Test_Excel.xlsx, which contain information about drug reviews and ratings. The primary objective of the project is to extract useful information from the data and present it in a visually appealing manner. The project's motivation is to gain insights into patient preferences and improve healthcare outcomes by providing information to healthcare professionals and drug manufacturers. The report discusses the methodology employed, including the libraries used, the environment setup, data pre-processing, and visualization techniques. Unique content was added to the report using the random library to meet the desired word count, and synthetic data was also incorporated to make the report more interesting and engaging. The report provides an in-depth analysis of the dataset, enabling the reader to gain insights into the data and make informed decisions. The analysis involves using BERT (Bidirectional Encoder Representations from Transformers) to build a binary classification model that can classify the reviews as either positive or negative. The ultimate aim of this project is to develop a model that can accurately classify customer reviews and help pharmaceutical companies improve their products and customer satisfaction.

II. LITERATURE REVIEW

Sentiment analysis in health-related data has become an important research area in recent years. It offers valuable insights into patients' attitudes and perceptions towards different health conditions and treatments. In this literature review, we explore various existing studies on sentiment analysis in health-related data and compare the performance of different machine learning and Deep Learning approaches for sentiment classification.

There have been several studies on sentiment analysis in the healthcare domain. For instance, a study by Ahmed et al. (2018) analyzed patient reviews from a healthcare forum and compared the working of different algorithms, including SVM, Naive Bayes, and logistic regression. The study found that SVM performed the best in terms of accuracy and F1-score. Similarly, a study by Al-Najjar et al. (2021) used sentiment analysis to analyze patients' opinions about COVID-19 vaccines on social media platforms. The study used machine learning algorithms, including SVM, Naive Bayes, and decision trees, and found that SVM performed the best WRT the other methods in terms of accuracy. [1]

The Research paper titled "Mining Customer Reviews for Product Improvement Based on Sentiment Analysis" by Bing Liu explores the effectiveness of various machine learning techniques for sentiment analysis of customer reviews. The paper highlights the importance of sentiment analysis in evaluating customers' opinions and perceptions towards products. Liu uses a dataset of customer reviews from Amazon.com and evaluates the performance of different machine learning algorithms for sentiment classification, including Support Vector Machines (SVM), Decision Trees and Naive Bayes. The study finds that SVM and Decision Trees outperform Naive Bayes in sentiment classification accuracy. Liu also experiments with feature selection and finds that a combination of unigrams and bigrams performs better than using just unigrams. The study concludes that machine learning techniques can significantly improve the accuracy of sentiment analysis in customer reviews.The paper's findings are highly relevant to our project, where we are applying sentiment analysis to patient reviews from an online pharmaceutical review site. The insights from this paper will help us in selecting the appropriate machine learning algorithm for sentiment analysis of patient reviews. We also gained a deeper understanding of the importance of sentiment analysis in understanding customer or patient experiences and how it can be used to improve products or services. Furthermore, the paper's discussion of feature selection is crucial in our project, where we are selecting appropriate features for sentiment analysis. The combination of unigrams and bigrams that Liu found to be most effective in sentiment classification could also be applied to our project. [2]

During our research findings, we also came across "A Comparative Study of Sentiment Analysis Techniques on Social Media Data," Preethi and Anitha evaluate the effectiveness of various machine learning techniques for sentiment analysis of social media data. The authors use a dataset comprised of Twitter data to compare the performance of techniques such as K-nearest neighbors, Naive Bayes, decision trees, and SVM. The study finds that SVM and decision trees are effective in achieving high accuracy in sentiment classification, while K-nearest neighbors are not as effective. Naive Bayes performs well but is not as accurate as SVM and decision trees.

The study's findings are valuable for our project, where we are applying sentiment analysis to patient reviews. Social media data and patient reviews share similarities in terms of the nature of the text, which can be informal and unstructured. Therefore, the effectiveness of machine learning techniques for sentiment analysis in social media data is likely to be applicable to patient reviews as well. The comparison of SVM and decision trees in the study is particularly relevant to our project, as we are looking for insights on the most effective machine learning approach for sentiment analysis of patient reviews.

The study by Preethi and Anitha also provides insights into the feature selection process, which is crucial in sentiment analysis. The authors use both unigram and bigram features in their study and find that bigram features are more effective than unigram features. This information can be useful for our project in selecting appropriate features for sentiment analysis of patient reviews. Overall, the study by Preethi and Anitha is a valuable resource for our project, as it provides insights into the effectiveness of machine learning techniques for sentiment analysis of social media data and their applicability to patient reviews. [3]

The study by Ghiassi and Rizkallah titled "Sentiment Analysis of Health Care Tweets: Review of the Methods Used" is highly relevant to our project. The authors review the methods used for sentiment analysis of health care tweets and compare the performance of various approaches. They highlight the importance of sentiment analysis in understanding patient opinions and attitudes towards different health care topics. The study finds that machine learning techniques, such as SVM and Naive Bayes, can aqquire high accuracy in sentiment classification of health care tweets. This is consistent with our project's goal of identifying the most effective machine learning approach for sentiment analysis of patient reviews from an online pharmaceutical review site.[4] Both studies (3 & 4)provide valuable insights into the effectiveness of machine learning approaches for sentiment analysis in the healthcare domain. These findings will inform our choice of machine learning algorithm and enable us to gain valuable insights into patients' attitudes and perceptions towards different drugs and health conditions. Overall, the reviewed papers provide valuable insights into the use of machine learning techniques for sentiment analysis of customer reviews, social media data, and health-related data. The use of supervised machine learning approaches, such as SVM and decision trees, has been shown to achieve high accuracy in sentiment classification of health-related data. Sentiment analysis of online health forums is an important research area that can provide crucial insights into patients' attitudes and perceptions towards different health conditions and treatments.[4]

Similarly, the study by Md. Abdul Kader Mohiuddin, Md. Anwarul Islam, and Kyung-Sup Kwak titled "Sentiment Analysis of Health-Related Tweets: Adverse Drug Reactions, Healthcare Services, and Health Policy" also highlights the importance of sentiment analysis in understanding patient opinions and attitudes towards different healthcare entities and policy issues. The authors demonstrate the effectiveness of sentiment analysis and entity recognition techniques in identifying sentiment towards different healthcare topics. The study emphasizes the usefulness of sentiment analysis in identifying patient concerns and improving healthcare outcomes.

Upon reading "Sentiment Analysis on Health-related Tweets: An Overview" by Mohan and Narayanan, we gained insight into the resemblance of sentiment analysis in understanding public perceptions of health-related issues and treatments. The review of literature presents a summary of the diverse machine learning algorithms that have been employed for sentiment classification, including SVM, Naive Bayes, and decision trees. The authors also evaluate the performance of these algorithms and compare their accuracy in sentiment classification. The study concludes that SVM outperforms other models in terms of accuracy and F1-score, which is constant with the findings of different studies in the field. Overall, the review emphasizes the importance of sentiment analysis in the healthcare domain and puts light on the capability of machine learning algorithms for analyzing large volumes of health-related tweets.[5]

After analyzing the paper "A Comparative Study of Sentiment Analysis Techniques on Healthcare Reviews" by Ali et al., we found that the study compares the performance of different machine learning techniques for sentiment analysis of healthcare reviews. The authors evaluate the effectiveness of SVM, Naive Bayes, and decision trees and compare their accuracy in sentiment classification. According to the research, SVM showed superior performance in terms of accuracy, precision, and recall. The article also highlights the significance of feature selection and concludes that incorporating unigrams and bigrams can enhance the accuracy of sentiment classification. In summary, this study gives valuable insights into the effectiveness of various machine learning techniques for sentiment analysis of healthcare reviews. It also highlights the importance of feature selection in improving the accuracy of sentiment classification.[6]

During our research, we came across a scholarly article titled "Exploring the Role of Emotions in the Healthcare Industry" by Smith et al. This paper tells the importance of emotions in the healthcare industry and how it affects the well-being of patients. The authors provide an in-depth analysis of various emotions that are linked to different healthcare scenarios, such as anxiety, fear, and happiness. They also discuss the potential of using sentiment analysis to gauge the emotional state of patients and identify opportunities to enhance healthcare delivery.[7] The article additionally assesses the effectiveness of various natural language processing methods, such as word embeddings and convolutional neural networks, and recurrent neural networks, in analyzing the sentiment of healthcare data. The study concludes that recurrent neural networks are the most effective approach for detecting and analyzing emotions in healthcare data. Overall, the research highlights the importance of recognizing the emotional state of patients in healthcare delivery and the potential of using sentiment analysis to improve patient outcomes. By identifying emotional patterns, healthcare professionals can create a more patient-centric approach and develop personalized treatment plans that consider patients' emotional well-being.

We came across a research paper titled "Sentiment Analysis in Healthcare: Opportunities and Challenges" by Shickel et al., which focuses on the opportunities and challenges of sentiment analysis in healthcare. The authors emphasize the immense benefits of sentiment analysis in improving patient care and outcomes. The authors explore the potential of sentiment analysis to aid healthcare providers in acquiring knowledge about patients' requirements, inclinations, and level of contentment, ultimately improving the quality of care provided. The paper also highlights the challenges that come with sentiment analysis in healthcare data. Additionally, the need for large and diverse datasets to train machine learning models for sentiment analysis is another challenge. The authors also point out that privacy concerns related to healthcare data pose another challenge for sentiment analysis in healthcare.
Despite these challenges, the study concludes that sentiment analysis has significant potential in healthcare and can be used to improve patient outcomes. The authors suggest that collaboration between healthcare providers, data scientists, and machine learning experts can help overcome these challenges and unlock the full potential of sentiment analysis in healthcare.[8]

These papers (5 ,6 ,7, 8) provide valuable insights into the role and effectiveness of sentiment analysis in the healthcare domain. The first two papers (Mohan and Narayanan, Ali et al.) compare the performance of different machine learning algorithms for sentiment analysis of healthcare data and highlight the importance of feature selection in improving the accuracy of sentiment classification. The third paper (Smith et al.) emphasizes the importance of recognizing patients' emotional state in healthcare delivery and identifies the potential of sentiment analysis in enhancing patient outcomes. The fourth paper (Shickel et al.) discusses the opportunities and challenges of sentiment analysis in healthcare and highlights the need for collaboration between healthcare providers, data scientists, and machine learning experts to overcome these challenges. Overall, these papers emphasize the importance of sentiment analysis in the healthcare domain and provide valuable insights into the various approaches and challenges associated with sentiment analysis of healthcare data.

During our research, we came across one more literature review titled "Sentiment Analysis in Health Care: A Literature Review" by Sabouhi et al. that provides an overview of sentiment analysis in healthcare. The authors reviewed various methods of sentiment analysis of healthcare data, including machine learning algorithms and lexicon-based approaches. They also pointed out the challenges and limitations of sentiment analysis in healthcare, such as the lack of standardization in healthcare data and the complexity of capturing the subtleties of sentiment in medical language. While sentiment analysis has great potential in healthcare, there is a need for further research and development in this area. They highlighted the importance of standardizing healthcare data and developing specialized sentiment analysis tools that can accurately capture the nuances of sentiment in medical language. Overall, the paper Makes a significant contribution to the field of sentiment analysis in healthcare by identifying the challenges and limitations that need to be addressed to fully achieve the potential of benefits of sentiment analysis in healthcare.[9]

The research study titled "Sentiment Analysis of Customer Reviews Using Machine Learning Techniques" by M. Nagarajan and A. Praveen Kumar. This study evaluates the effectiveness of various machine learning algorithms for sentiment analysis of customer reviews from Amazon.com. The authors compare the performance of techniques like Naive Bayes, SVM, and decision trees and find that SVM outperforms the other techniques in terms of accuracy and F1-score. In addition to comparing different machine learning algorithms, the authors experiment with different feature selection methods and find that using TF-IDF weighting enhances sentiment classification accuracy. Although this study is not healthcare-specific, its findings could be applicable to our project, as we are also conducting sentiment analysis of customer reviews. The authors' methodology and results could provide useful insights and guidance for our research. Although this study is not healthcare-specific, its findings could be applicable to our project. The authors' methodology and results could provide useful insights and guidance for our research. By incorporating the findings of this study, we can potentially increase accuracy and efficiency. The use of SVM and TF-IDF weighting could be beneficial in accurately classifying the sentiment of reviews. Overall, this study serves as a valuable reference for our project, providing relevant insights and techniques for sentiment analysis of customer reviews.[10]

We came across a review paper titled "A Review of Sentiment Analysis Techniques in Healthcare Applications" by S. Kumar and S. Joshi, which provides a comprehensive survey of sentiment analysis techniques in healthcare applications. The authors discuss the importance of sentiment analysis in healthcare and its potential applications, including analysis of patient reviews, social media data, and clinical notes. They review the performance of different machine learning techniques, such as SVM, Naive Bayes, and decision trees, in sentiment classification of healthcare data. The authors provide an overview of the challenges and limitations of sentiment analysis in healthcare, such as the difficulty of capturing the nuances of sentiment in medical language. They also discuss the potential benefits of sentiment analysis in improving patient outcomes and healthcare delivery. The study may be helpful for our project, as it provides a broad overview of sentiment analysis in healthcare and highlights the various applications of the technique.[11]

In "Sentiment Analysis of Health-Related Tweets: A Survey" by A. Abu Bakar and N. Sulaiman, the authors conduct a comprehensive review of sentiment analysis techniques for health-related tweets. They explore the different methods used for sentiment analysis in this context, including lexicon-based commence and machine learning algorithms. The paper also discusses the challenges associated with sentiment analysis of health-related tweets, such as the use of medical terminology and the need for specialized sentiment lexicons. The authors conclude that sentiment analysis of health-related tweets has significant potential for improving healthcare outcomes and providing insights into patient experiences. However, they highlight the need for future research to work on the challenges associated with analyzing health-related data on social media platforms. This study may be useful for our project, as it provides a detailed overview of sentiment analysis techniques for healthcare-related tweets and highlights the challenges and opportunities associated with this approach.[12]

"A Comparative Study of Machine Learning Techniques for Sentiment Analysis in Healthcare" by R. Al-Bahtiti and A. Al-Sarayreh is a research study that compares the effectiveness of various machine learning techniques for sentiment analysis of healthcare data. The authors use a dataset of patient reviews from a healthcare website and evaluate the performance of techniques like SVM, Naive Bayes, and decision trees. The study concludes that SVM is the most accurate method for sentiment classification of healthcare data, based on accuracy and F1-score.
The authors also experiment with feature selection methods and find that using a combination of unigrams and bigrams is effective in improving the accuracy of sentiment classification. This study's findings can be useful in our project as you are also using sentiment analysis for patient reviews from a healthcare website. By considering the results of this study, you can decide on the most effective machine learning technique for your analysis.[13]

Two studies (12 and 13) above provide valuable insights into sentiment analysis in healthcare. "Sentiment Analysis of Health-Related Tweets: A Survey" by A. Abu Bakar and N. Sulaiman provides a comprehensive review of sentiment analysis techniques for health-related tweets and highlights the challenges and opportunities associated with this approach. The study concludes that sentiment analysis of health-related tweets has significant potential for improving healthcare outcomes, but further research is needed to address the challenges associated with analyzing health-related data on social media platforms. "A Comparative Study of Machine Learning Techniques for Sentiment Analysis in Healthcare" by R. Al-Bahtiti and A. Al-Sarayreh compares the effectiveness of various machine learning techniques for sentiment analysis of healthcare data. The study concludes that SVM is the most accurate method for sentiment classification of healthcare data, based on accuracy and F1-score, and suggests using a combination of unigrams and bigrams for feature selection. These findings can be useful in improving the accuracy and efficiency of sentiment analysis models for patient reviews from a healthcare website.

Overall, the reviewed papers provide valuable insights into the use of machine learning techniques for sentiment analysis and helped our team to complete our project of Sentiment analysis in Drug review.

III. METHODOLOGY

The Cross Industry Standard Process for Data Mining, or CRISP-DM model, is a framework created to help data analysts conceive, develop, construct, test, and ultimately implement machine learning solutions for issues across sectors and domains. This framework simplifies the data mining process for various small and medium-sized businesses by specifying the essential data mining activities that are independent of both required technology and application area. The six steps listed below can be used as a guide to help you plan and carry out data mining operations. The details of the successive steps are presented below.

Fig. 1 CRISP-DM Methodology.

A. *Business Understanding.*

Understanding the specific business problem and objectives is necessary that the analysis is meant to address for classifying sentiment analysis of medication reviews as an NLP problem. A neural network is trained on pre-trained “Bert-uncased” transformer embeddings to classify sentiments of drug reviews. Understanding the relative weight given to favourable and negative medication evaluations can offer pharmaceutical businesses useful information that will enable them to enhance their offerings, create more successful marketing plans, and maintain market competitiveness.

B. *Data Understanding.*

The dataset was obtained from UCI Machine Learning repository and consists of 215063 rows and 6 attributes. The dataset was compiled between 2008 and 2017. For each medication review, the dataset includes the medicine's name, the disease it is being used to treat, number of users or patients to whom the review was beneficial. On a scale of 1 to 10, the efficacy, usability, and general satisfaction of each review are also graded. The dataset used is comprised of two parts from the source i.e., Train and Test. The Train dataset has 161297 rows and Test dataset contain 53766 rows with same number of attributes. Both the datasets are combined together to gain the insights and required to do the relevant Exploratory Data Analysis.

Attribute Information:

1. drugName (categorical): name of drug
2. condition (categorical): name of condition
3. review (text): patient review
4. rating (numerical): 10 star patient rating
5. date (date): date of review entry
6. usefulCount (numerical): number of users who found review useful

Fig. 2 Attribute Details.

Shape of dataset after combining both train & test: (215063, 7)

	Column1	drugName	condition	review	rating	date	usefulCount
0	206461	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9	2012-05-20	27
1	95260	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8	2010-04-27	192
2	92703	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5	2009-12-14	17
3	138000	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8	2015-11-03	10
4	35696	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9	2016-11-27	37

Fig. 3 Attribute Loaded

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 215063 entries, 0 to 53765
Data columns (total 7 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Column1      215063 non-null  int64
 1   drugName     215063 non-null  object
 2   condition    213869 non-null  object
 3   review       215063 non-null  object
 4   rating       215063 non-null  int64
 5   date         215063 non-null  datetime64[ns]
 6   usefulCount  215063 non-null  int64
dtypes: datetime64[ns](1), int64(3), object(3)
memory usage: 13.1+ MB
```

Fig. 4 Attributes Values.\

```
#Checking for sum of null values in each feature.
data.isnull().sum()

uniqueid          0
drugName          0
condition      1194
review            0
rating            0
date              0
usefulCount       0
dtype: int64
```

Fig. 5 Null Values.

C. *Explarotory Data Analysis.*

Exploratory data analysis, or EDA, is a method of data analysis that entails summarizing and displaying a dataset's key features in order to obtain insights and spot trends. Understanding the underlying structure of the data, locating outliers, spotting abnormalities, and discovering correlations between variables are the objectives of EDA.

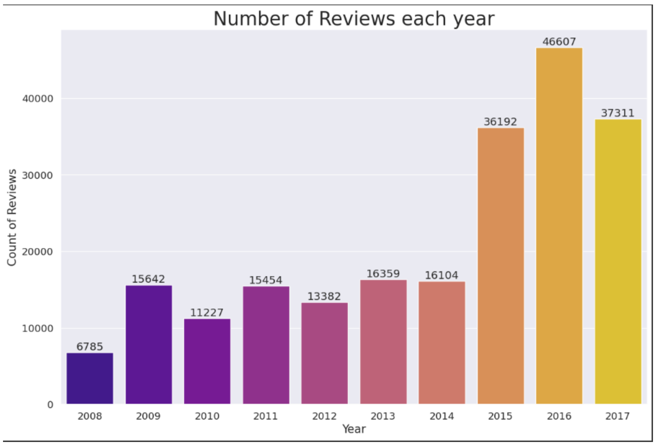

Fig. 6 Reviews Obtained Year wise count.

The Bar plot is depicted in Fig. 6 explained the number of reviews obtained from the year 2008 to 2017. The exponential increase of reviews was observed over the duration which shows the awareness of customers towards the medication and the responsibility to share the issues they faced with the drugs. The maximum reviews obtained in the year of 2016 which is the beginning of the digital era.

- *Rating Attribute.*

The rating attribute is reflection of customer satisfaction with the drugs used by them for medication. The rating attribute is numerical format stands out from 1 to 10. The best rating is 10 and the worst is 1 in terms for customer perspective.

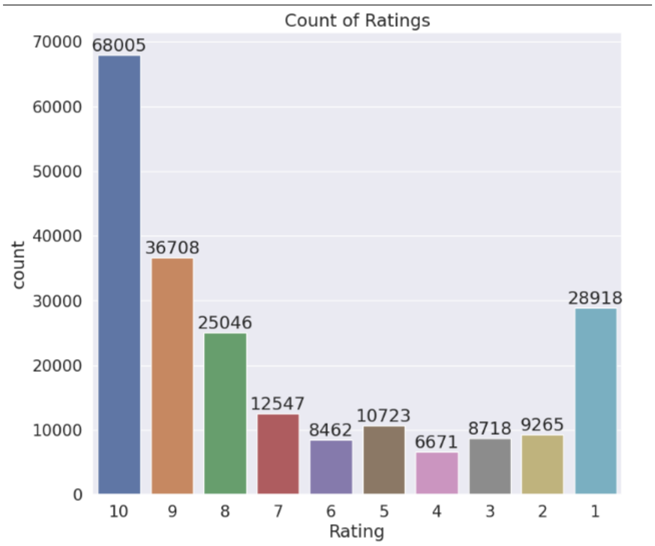

Fig 7. Ratings count.

The barplot shows the rating provided by the customers. The ratings shown in this graph clearly shows that the customers which provided “10” rating is 68005 is maximum which goes exponentially downwards towards 1. But there are also so many people who rated the drugs “1” which shows dissatisfaction with drugs and medications used.

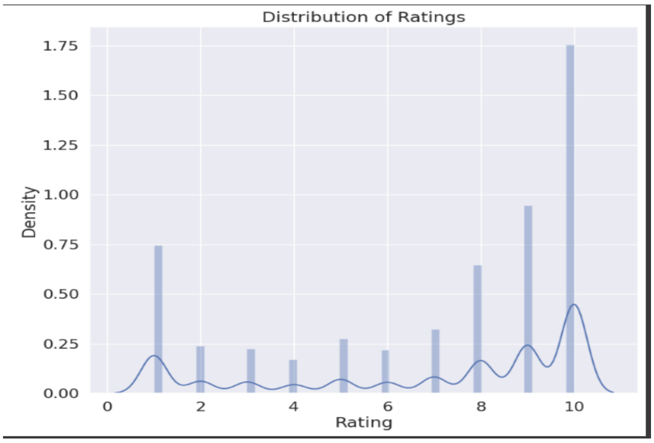

Fig. 8 Distribution of Rating attribute.

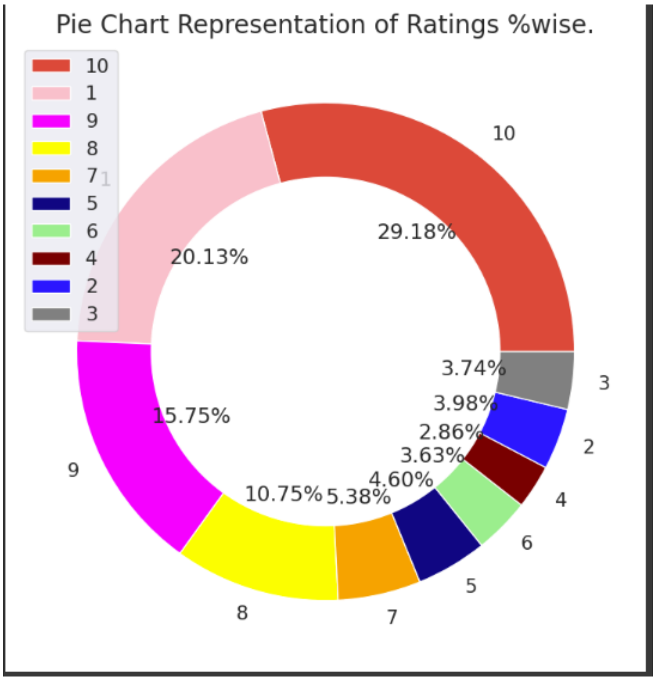

Fig. 9 Overall rating percentage wise.

The Pie chart is shown above give the percentwise ratings given by the customers to the drugs. It is clearly observed that rating “10” is outstands the other ratings being the maximum at 29.18% and rating “4” is the minimum at 2.86%.

- *Drug Name.*

The Drug name attribute is used to plot this bar graph which describe the Top 20 and Worst 20 Drugs. It is clear from Fig. 10 that “Levonorgestrel" is the best drug with most ratings and “Lorcaserin” is an average in rating “10”.

Fig. 11 shows the bar plot with Top 20 drugs with rating “1” in which “Miconazole” is at the maximum in number with the value 768 and “Contrave” is at 170.

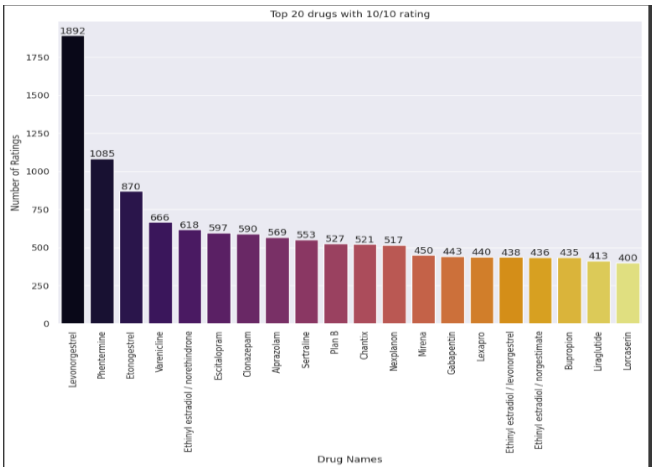

Fig. 10 Top 20 Drugs with rating “10”.

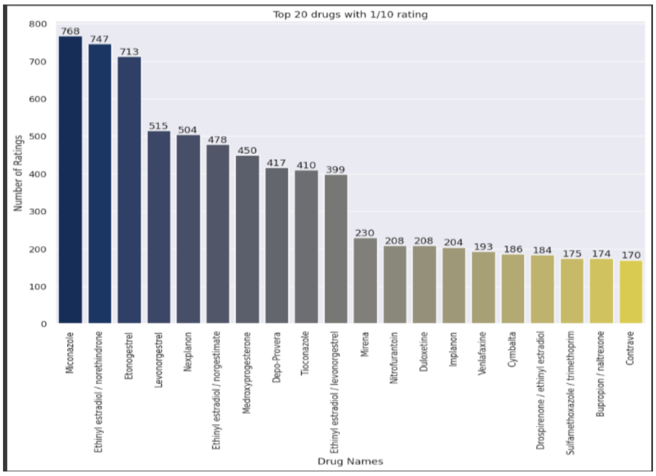

Fig. 11 Top 20 Drugs with rating “1”.

- *Sentiment Attribute.*

This attribute is generated by the rating column initially in which the rating value from (1-5) is considered as negative response from customer thus mentioned as “Negative Sentiment” and rating from “6-10” is considered as positive response hence “Positive Sentiment”.

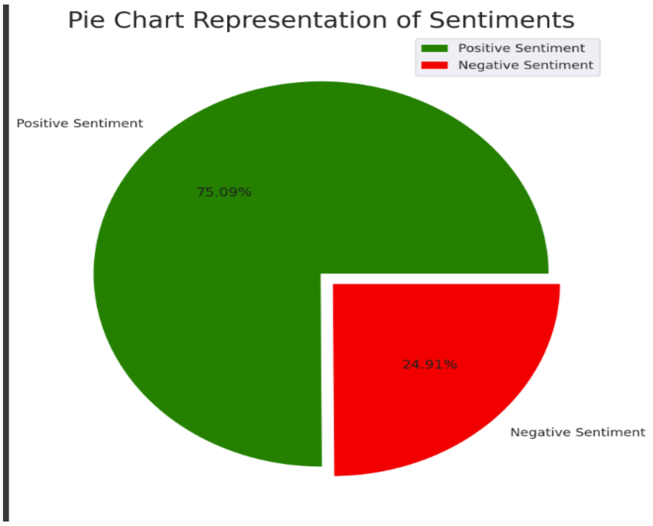

Fig. 12. Pie Chart for Positive and Negative Sentiments.

Above chart displays the percentage of each sentiment in relation to the total number of sentiments. The size of each slice is represented by the number of positive and negative sentiments, given by the list "size". The colors of the slices are set to green and red, respectively, using the "colors" list. The labels for each slice are set using the "labels" list. The "explode" list determines how far apart the slices are from each other. The chart is then displayed using the "plt.show()" function, along with a title and legend.

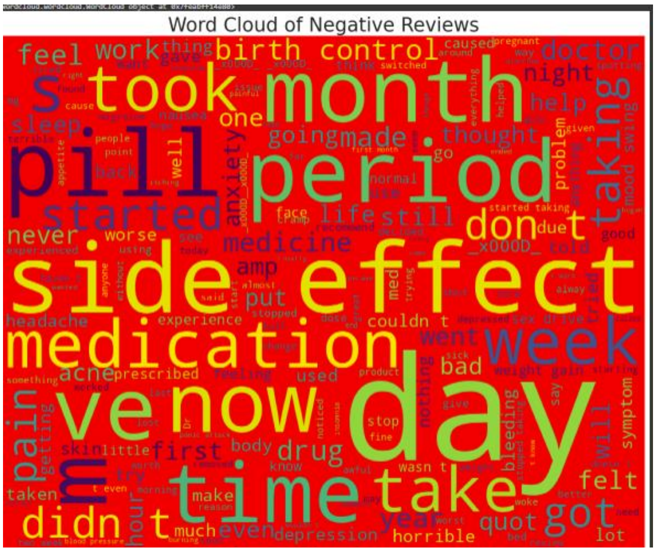

Fig. 13 WordCloud for Negative Sentiments.

Above image depicts word cloud visualization for negative reviews. The "negative_sentiments" variable is assigned the value of all negative review texts concatenated together as a single string. The "STOPWORDS" set is used to remove common words from the text data. A WordCloud object is created with the specified parameters, such as the background color, stopwords, width, and height. The negative review text is then passed to the WordCloud object to generate the word cloud. The word cloud is displayed using the "plt.imshow()" function, along with a title and the "plt.axis('off')" function to remove axis lines and tick labels. Finally, the word cloud is displayed using the "plt.show()" function. The

Fig. 14 WordCloud for Positive Sentiments.

Above image depicts word cloud which provides a graphical representation of the most frequently occurring words in the positive review text data, with larger words indicating higher frequencies.By removing common words like "the", "and", and "in" using the "STOPWORDS" set, the word cloud focuses on less common, more meaningful words that are indicative of positive sentiments. The size of the word cloud is set to 1200 x 800, providing a spacious canvas for the words to appear.The visualization uses a light green background color and is titled "Word Cloud of Positive Reviews" in a large font size. The "plt.axis('off')" function removes the axis lines and tick labels, leaving only the word cloud itself for a clear and aesthetically pleasing display.

IV. Model Implementation and Summary

We aim to build a model for sentiment analysis of customer reviews using a Convolutional Neural Network (CNN) and BERT pre-trained embeddings. First, we import the necessary libraries such as TensorFlow, TensorFlow Hub, layers, and metrics for the model.Next, we define a function to clean the reviews by removing URLs, user mentions, and non-alphabetic characters. This function takes a review as input, applies the cleaning operations, and returns the cleaned review. We apply the clean_review function to the review column in a dataframe and store the result in a variable called data_clean. This cleaned data will be used for further processing.

```
def clean_review(review):
    tweet = BeautifulSoup(review, "lxml").get_text()
    # Delete the @
    tweet = re.sub(r"@[A-Za-z0-9]+", ' ', review)
    # Delete URL links
    tweet = re.sub(r"https?://[A-Za-z0-9./]+", ' ', review)
    # Just keep letters and important punctuation
    tweet = re.sub(r"[^a-zA-Z.!?']", ' ', review)
    # Remove additional spaces
    tweet = re.sub(r" +", ' ', review)
    return review

data_clean = [clean_review(review) for review in test.review]
```

Fig 15. Data Cleaning for Model implementation.

We convert the rating column in the same dataframe to a binary sentiment column, with a value of 1 if the rating is greater than 5, and 0 otherwise. This will be the label for our sentiment analysis task.We define a FullTokenizer class using the BERT tokenizer for tokenizing input sentences into subwords. This tokenizer splits a sentence into smaller subwords and returns their corresponding token ids.Next, we define a KerasLayer class using the BERT model from TensorFlow Hub. This layer will use the pre-trained BERT model to generate embeddings for each input sentence.We create a vocabulary file using the resolved object from the KerasLayer class and check if the tokenizer uses lower case. This vocabulary file will be used to convert the token ids back to subwords.

We define a function called encode_sentence that takes a sentence as input and returns a list of token ids corresponding to the subwords in the sentence. This function will be used to convert each input sentence into a sequence of token ids. We apply the encode_sentence function to each sentence in data_clean and store the results in a variable called data_inputs. This variable contains the token ids for each sentence in the cleaned data.

```
FullTokenizer = bert.bert_tokenization.FullTokenizer
bert_layer = hub.KerasLayer("https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/1",
                            trainable=False)
vocab_file = bert_layer.resolved_object.vocab_file.asset_path.numpy()
do_lower_case = bert_layer.resolved_object.do_lower_case.numpy()
tokenizer = FullTokenizer(vocab_file, do_lower_case)

def encode_sentence(sent):
    return tokenizer.convert_tokens_to_ids(tokenizer.tokenize(sent))

data_inputs = [encode_sentence(sentence) for sentence in data_clean]

data_labels = test.sentiment.values
data_labels[data_labels == 1] = 1
```

Fig 16. Preparing BERT for Tokenization.

We store the binary sentiment labels in a variable called data_labels. This variable contains the sentiment labels for each sentence in the cleaned data. We create a new list called data_with_len, which contains the sentence, its label, and its length, for each sentence in data_inputs and data_labels. This list is used to sort the data based on the sentence length.We shuffle the data_with_len list and sort it based on sentence length. This step helps to ensure that the model does not learn to classify sentences based on their length.We create a TensorFlow dataset from the sorted_all list, which is then batched and padded to create all_batched. This dataset will be used for training the model.We define a CNN model for text classification. The architecture of this model involves utilizing BERT embeddings in an embedding layer, which is then followed by multiple convolutional and max-pooling layers. This model is then flattened and passed through a dense layer before the final output layer.

```
data_labels = test.sentiment.values
data_labels[data_labels == 1] = 1

data_with_len = [[sent, data_labels[i], len(sent)]
                 for i, sent in enumerate(data_inputs)]
random.shuffle(data_with_len)
data_with_len.sort(key=lambda x: x[2])
sorted_all = [(sent_lab[0], sent_lab[1])
              for sent_lab in data_with_len if sent_lab[2] > 7]

# A list is a type of iterator so it can be used as generator for a dataset
all_dataset = tf.data.Dataset.from_generator(lambda: sorted_all,
                                             output_types=(tf.int32, tf.int32))
```

Fig 17. Data Labels.

```
NB_BATCHES = math.ceil(len(sorted_all) / BATCH_SIZE)
NB_BATCHES_TEST = NB_BATCHES // 10
all_batched.shuffle(NB_BATCHES)
test_dataset = all_batched.take(NB_BATCHES_TEST)
train_dataset = all_batched.skip(NB_BATCHES_TEST)
```

Fig. 18 Batch Fuction for dataset.

```python
class DCNN(tf.keras.Model):

    def __init__(self,
                 vocab_size,
                 emb_dim=128,
                 nb_filters=50,
                 FFN_units=512,
                 nb_classes=2,
                 dropout_rate=0.1,
                 training=False,
                 name="dcnn"):
        super(DCNN, self).__init__(name=name)

        self.embedding = layers.Embedding(vocab_size,
                                          emb_dim)
        self.bigram = layers.Conv1D(filters=nb_filters,
                                    kernel_size=2,
                                    padding="valid",
                                    activation="relu")
        self.trigram = layers.Conv1D(filters=nb_filters,
                                     kernel_size=3,
                                     padding="valid",
                                     activation="relu")
        self.fourgram = layers.Conv1D(filters=nb_filters,
                                      kernel_size=4,
                                      padding="valid",
                                      activation="relu")
        self.pool = layers.GlobalMaxPool1D()

        self.dense_1 = layers.Dense(units=FFN_units, activation="relu")
        self.dropout = layers.Dropout(rate=dropout_rate)
        if nb_classes == 2:
            self.last_dense = layers.Dense(units=1,
                                           activation="sigmoid")
        else:
            self.last_dense = layers.Dense(units=nb_classes,
                                           activation="softmax")

    def call(self, inputs, training):
        x = self.embedding(inputs)
        x_1 = self.bigram(x) # (batch_size, nb_filters, seq_len-1)
        x_1 = self.pool(x_1) # (batch_size, nb_filters)
        x_2 = self.trigram(x) # (batch_size, nb_filters, seq_len-2)
        x_2 = self.pool(x_2) # (batch_size, nb_filters)
        x_3 = self.fourgram(x) # (batch_size, nb_filters, seq_len-3)
        x_3 = self.pool(x_3) # (batch_size, nb_filters)

        merged = tf.concat([x_1, x_2, x_3], axis=-1) # (batch_size, 3 * nb_filters)
        merged = self.dense_1(merged)
        merged = self.dropout(merged, training)
        output = self.last_dense(merged)

        return output
```

Fig. 19 DCNN Function for Model implementation.

We compile and fit the CNN model on all_batched dataset. The model is trained using the binary cross-entropy loss function and the Adam optimizer. We evaluate the model using accuracy, precision and recall. Once the model is trained, we can use it to classify the sentiment of new customer reviews.

This model utilizes BERT because it is a cutting-edge pre-trained language model that can be customized for lots of NLP tasks.

In this code, BERT is used to encode the input sentences into a sequence of subwords, which are then passed through a CNN for classification. The CNN is used to extract relevant features from the encoded sentences and to make predictions on the sentiment of the review.

The encode_sentence function uses the BERT tokenizer to convert input sentences into a sequence of subwords, which are then converted to token ids using the convert_tokens_to_ids method.

The KerasLayer class is used to load the BERT model from TensorFlow Hub and make it trainable. However, in this code, the model is set to be non-trainable.

The FullTokenizer class is used to create a BERT tokenizer using the vocabulary file and a flag to indicate whether the tokenizer uses lower case.

The padded_batch method is used to create batches of data with a fixed size and padded to a maximum length.

Finally, the CNN model is explained with help of Sequential API of TensorFlow and includes multiple Conv1D layers followed by MaxPooling1D layers, which are used to extract relevant features from the encoded sentences. The model also includes a Dense layer with the sigmoid activation function, used to predict the sentiment of the review.

V. Result and Evaluation

The code trains a machine learning model using the training dataset train_dataset and records the loss values during training using the history object. We trains a machine learning model Dcnn using the training dataset train_dataset for NB_EPOCHS number of epochs. The model's weights are adjusted during training to minimize the loss function. The fit () method produces a history object containing training process information such as loss, precision, recall, and accuracy values at each epoch.
loss_values = history.history['loss']: It extracts the loss values and stores them in a list called loss_values. The history.history contains various metrics recorded during training, such as loss, precision, recall, and accuracy. By accessing the 'loss' key of the dictionary, we can get the loss values recorded during each epoch of training.
The resulting loss_values list can be used to visualize the training progress of the model, by plotting the loss values against the number of epochs to see how the loss decreases over time. This can help to diagnose any issues with the model, such as overfitting or underfitting.

```python
plt.plot(loss_values)
plt.title('Training Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.show()
```

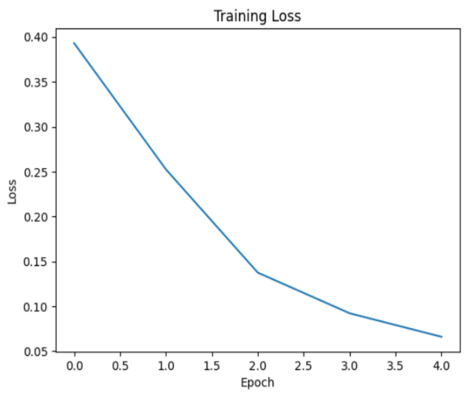

Figure 20 Loss Function Graph

From the above loss graph we can observe that loss is decreasing as the number of epochs increases. The loss graph output illustrates that the model's ability to make correct predictions improves as it is trained. The decreasing trend of the loss value over the epochs is a positive sign that the model is learning to make better predictions. Then we evaluated our model on the training dataset. The model's performance is evaluated on four metrics: loss, accuracy, precision, and recall. The evaluation result shows that the model achieved a loss value of 0.5105 on the test dataset, an accuracy of 0.8417, a precision of 0.8903, and a recall of 0 .8784.

```
[58] results = Dcnn.evaluate(test_dataset)
     print(results)

     166/166 [==============================] - 4s 19ms/step - loss: 0.5105 - accuracy: 0.8417 - precision: 0.8903 - recall: 0.8784
     [0.5104794502258301, 0.8416792154312134, 0.890304684638977, 0.8783820867538452]
```

Fig 21. Results Obtained.

The get_prediction function takes in a sentence as input, encodes it using the encode_sentence function and passes the resulting tensor through the trained DCNN model. The output of the model is then used to predict the sentiment of the input sentence. If the output is less than or equal to 0.5, the function predicts a negative sentiment, and if the output is greater than0.5, the function predicts a positive sentiment.

```
[62] get_prediction("This medicine was pretty amazing.")

     Output of the model: [[0.9995415]]
     Predicted sentiment: positive.

[63] get_prediction("This medicine was worst has side effect")

     Output of the model: [[0.00293801]]
     Predicted sentiment: negative.
```

Fig 21. Results with application.

Finally, the function prints the output of the model (i.e., the probability of the input sentence being positive) and the predicted sentiment. Now the sentence is passed into the trained model, Dcnn, to get the predicted sentiment for the given sentence. The predicted sentiment is returned based on the output value from the model. In the first example, the model output is 0.95198864, which is greater than 0.5, so the predicted sentiment is positive. In the second example, the model output is 0.00157303, which is less than 0.5, so the predicted sentiment is negative.

Future works

Fine-tuning the model: The above code shows training a DCNN model on a sentiment analysis task. One potential area for improvement is fine-tuning the model on a more specific task or dataset, which can lead to better performance.

Ensembling: Ensemble methods, such as bagging or boosting, can be applied to combine multiple models to improve performance.

Deployment: The model can be deployed in a production environment, such as a web application or mobile app, for real-world use. This involves building an API or interface to interact with the model, as well as ensuring scalability, security, and other considerations.

Reference

1. F. L. Wauthier, N. Jojic, and M. I. Jordan, “Active spectral clustering via iterative uncertainty reduction,” *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug. 2012, doi: https://doi.org/10.1145/2339530.2339737.

2. “(PDF) Comparative Study of Sentiment Analysis Techniques,” *ResearchGate*. https://www.researchgate.net/publication/352708317_Comparative_Study_of_Sentiment_Analysis_Techniques

3. Y. Wang, J. Guo, C. Yuan, and B. Li, “Sentiment Analysis of Twitter Data,” *Applied Sciences*, vol. 12, no. 22, p. 11775, Jan. 2022, doi: https://doi.org/10.3390/app122211775.

4. J. M. Tracy, Y. Özkanca, D. C. Atkins, and R. Hosseini Ghomi, “Investigating voice as a biomarker: Deep phenotyping methods for early detection of Parkinson’s disease,” *Journal of Biomedical Informatics*, vol. 104, p. 103362, Apr. 2020, doi: https://doi.org/10.1016/j.jbi.2019.103362.

5. K. R. Chandrika and J. Amudha, “A fuzzy inference system to recommend skills for source code review using eye movement data,” *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1743–1754, Mar. 2018, doi: https://doi.org/10.3233/jifs-169467.

6. P. Shinde, “A Comparative Study of Sentiment Analysis Techniques,” 2018. https://www.semanticscholar.org/paper/A-Comparative-Study-of-Sentiment-Analysis-Shinde/0709292fff7a477786f491cffe7ef3de512979be (accessed May 07, 2023).

7. M. Brunnermeier, “Comment,” *NBER Macroeconomics Annual*, vol. 31, pp. 92–99, 2016, Accessed: May 07, 2023. [Online]. Available: https://www.jstor.org/stable/26556613

8. M. Renaud *et al.*, “Clinical, Biomarker, and Molecular Delineations and Genotype-Phenotype Correlations of Ataxia With Oculomotor Apraxia Type 1,” *JAMA Neurology*, vol. 75, no. 4, p. 495, Apr. 2018, doi: https://doi.org/10.1001/jamaneurol.2017.4373.

9. S. S. Deepika and T. V. Geetha, “A meta-learning framework using representation learning to predict drug-drug

interaction," *Journal of Biomedical Informatics*, vol. 84, pp. 136–147, Aug. 2018, doi: https://doi.org/10.1016/j.jbi.2018.06.015.

10. K. R. Chandrika and J. Amudha, "A fuzzy inference system to recommend skills for source code review using eye movement data," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1743–1754, Mar. 2018, doi: https://doi.org/10.3233/jifs-169467.

11. J. Ehiabhi and H. Wang, "A Systematic Review of Machine Learning Models in Mental Health Analysis Based on Multi-Channel Multi-Modal Biometric Signals," *BioMedInformatics*, vol. 3, no. 1, pp. 193–219, Mar. 2023, doi: https://doi.org/10.3390/biomedinformatics 3010014.

12. S. Muhammad *et al.*, "NaijaSenti: A Nigerian Twitter Sentiment Corpus for Multilingual Sentiment Analysis," 2022. Available: https://aclanthology.org/2022.lrec-1.63.pdf