Text Based Sentiment Analysis Using NLP

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Information Technology

by

Manglam Paliwal 20208078
M Pooja 20208077
Nayan Lohiya 20208083
Ritik Sharma 20208108

to the

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD, PRAYAGRAJ
May, 2023

UNDERTAKING

I declare that the work presented in this report titled "**Text Based Sentiment Analysis Using NLP**", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, for the award of the Bachelor of Technology degree in **Information Technology**, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. If this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

May, 2023
Prayagraj

_____

Manglam Paliwal 20208078
M Pooja 20208077
Nayan Lohiya 20208083
Ritik Sharma 20208108

CERTIFICATE

Certified that the work contained in the report titled “Text Based Sentiment Analysis Using NLP”, by
Manglam Paliwal 20208078
M Pooja 20208077
Nayan Lohiya 20208083
Ritik Sharma 20208108
has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

_____

Dr. Dharmendra Kumar Yadav
Computer Science and Engineering Dept.
M.N.N.I.T. Allahabad

May, 2023
Prayagraj

Acknowledgements

I would like to express my sincere gratitude to Dr Dhamendra Kumar Yadav, our mentor, for his invaluable guidance and support throughout the completion of this report. His expertise in the subject matter and constant encouragement and feedback have been instrumental in shaping this report into its final form.

I would also like to extend my heartfelt thanks to the Motilal Nehru National Institute Of Technology Allahabad and the Government of India for providing the necessary resources and opportunities that enabled me to undertake this project.

Lastly, I would like to express my appreciation to all those who have contributed in some way to the completion of this report. Their assistance and encouragement have been vital to my success, and I am grateful for their support.

CONTENTS

1. Introduction

Text-based sentiment analysis is a technique used to automatically analyze and identify the emotional tone or attitude expressed in a piece of text, such as reviews, social media posts, or news articles. The process involves using natural language processing and machine learning algorithms to classify the text as either positive, negative, or neutral based on the language used and the context in which it appears. Sentiment analysis has numerous applications in fields such as marketing, customer service, and public opinion analysis. It allows organizations to gain insights into customer satisfaction, and sentiment trends and identify potential issues or opportunities for improvement.

Sentiment analysis has various applications across various industries, including marketing, social media monitoring, customer service, product development, and political analysis. It is commonly used to track brand reputation, measure customer satisfaction, analyze social media sentiment, and gauge public opinion on various issues. Sentiment analysis can also be used to identify emerging trends and patterns in customer behavior and preferences, helping organizations to make data-driven decisions.

1.1 Motivation

Deep learning is a subset of machine learning that uses artificial neural networks to analyze and extract patterns from large datasets. Deep learning has become increasingly popular in recent years due to its ability to process and analyze complex data, including natural language text. When applied to sentiment analysis, deep learning can help to improve the accuracy and efficiency of sentiment analysis algorithms, making it an exciting area of research and development.

Here are some reasons why someone might be motivated to undertake a project on sentiment analysis using deep learning:

Improved Accuracy: One of the main advantages of using deep learning for sentiment analysis is that it can lead to improved accuracy. Deep learning algorithms can analyze and learn from large amounts of data, allowing them to detect patterns and nuances in language that might be missed by traditional sentiment analysis methods.

Research: Sentiment analysis using deep learning is a rapidly evolving field of research, with new techniques and algorithms being developed all the time. Undertaking a project in this area can be an opportunity to contribute to the field and advance our understanding of sentiment analysis and deep learning.

Overall, the motivation for undertaking a project on sentiment analysis using deep learning is the potential for improved accuracy, flexibility, and real-time analysis. Additionally, sentiment analysis using deep learning is an exciting area of research with many opportunities for innovation and contribution to the field.

2. Proposed Work

- We have studied and implemented pre-trained models used for labeling and check their accuracy of them.

- We have developed a deep learning model for sentiment analysis for training and testing using a labeled dataset which gives better accuracy and F1-Score among pretrained model.

3. Experimental Setup

3.1 Technologies Used

Programs and tools that were very crucial for our project are:-

3.1.1 Python

Python is a high-level, interpreted programming language that is widely used in a variety of domains, including web development, data analysis, machine learning, and scientific computing. Python's popularity is largely due to its simplicity, readability, and versatility. Python has a vast collection of libraries, frameworks, and tools, making it a popular choice among developers. Some famous Python libraries include NumPy for scientific computing, Pandas for data manipulation, matplotlib for data visualization, Scikit-learn for machine learning, Django for web development, and Flask for building web applications. These libraries enable developers to write powerful, efficient, and scalable code with minimal effort, making Python a go-to language for building complex and data-intensive applications.

3.1.2 Jupyter Notebook

Jupyter Notebook is an open-source web application that allows users to create and share interactive data science and machine learning documents. Jupyter Notebook enables users to write and run code, visualize data, and document their work all in one place. It supports a wide range of programming languages, including Python, R, and Julia. Jupyter Notebook is widely used in the scientific and data analysis communities. It allows users to easily experiment and iterate on their code, analyze and visualize data, and document their findings. Jupyter Notebook has also become popular for teaching and learning data science and programming due to its interactive and user-friendly interface.

3.1.3 Scikit Learn

Scikit-learn is a popular open-source machine-learning library for Python. It provides a wide range of tools for data preprocessing, feature extraction, supervised and unsupervised learning, model evaluation, and data visualization. Scikit-learn is designed to be user-friendly, easy to use, and accessible to both novice and advanced users. It supports various machine learning algorithms, including classification, regression, clustering, and dimensionality reduction. Scikit-learn is widely used in academia and industry for various applications, including image and speech recognition, natural language processing, recommendation systems, and fraud detection. Its powerful and flexible API makes it a go-to tool for many machine learning practitioners and researchers.

3.1.4 Numpy and Pandas

NumPy and Pandas are the most popular Python libraries for data manipulation, analysis, and computation. NumPy provides a powerful array object and a collection of mathematical functions for working with large multi-dimensional arrays and matrices efficiently. It enables users to perform complex numerical operations such as linear algebra, Fourier transforms, and random number generation with ease. On the other hand, Pandas provides high-performance data structures and tools for data manipulation and analysis, such as data cleaning, aggregation, and transformation. It makes working with data in Python more convenient and efficient. NumPy and Pandas are widely used in various fields, including data science, finance, engineering, and physics, to name a few. Their efficient and powerful APIs make them essential tools in any data scientist's toolkit.

3.1.5 Tensorflow

TensorFlow is a free open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but focuses on training and inference of deep neural networks. TensorFlow's APIs use Keras to allow users to make their own machine-learning models. The Google Brain team developed TensorFlow for internal Google use in research and production.

3.1.6 Keras

It was developed by one of the Google engineers, Francois Chollet. Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. The tf.keras.datasets module provides a few toy datasets (already vectorized, in Numpy format) that can be used for debugging a model or creating simple code examples.

3.2 About Dataset

After exploring a plethora of data sets ranging from labeled to unlabelled, we have settled upon the **Movie Reviews dataset** with around 25000 reviews along with their sentiments (i.e., negative, neutral, positive), and **sentiment140 dataset** and **US Airline Sentiment** dataset from Kaggle, a trustable source of datasets.

3.3 Models Used

a) **Deep Learning Model**

The Deep Learning Model used is of 4 layers, containing following layers -:

keras.layers.Embedding(10000, 16, input_length=500)

keras.layers.GlobalAveragePooling1D()

keras.layers.Dense(16, activation='relu')

keras.layers.Dense(1, activation='sigmoid')

b) SVM

c) Logistic Regression

d) LSTM

4. Project Implementation

Project started with us studying and learning about sentiment analysis. And then for implementation, we started learning about Python language and jupyter notebook tool.

The following steps were followed to implement the project starting from data extraction to training and testing in a sequential manner as follows:

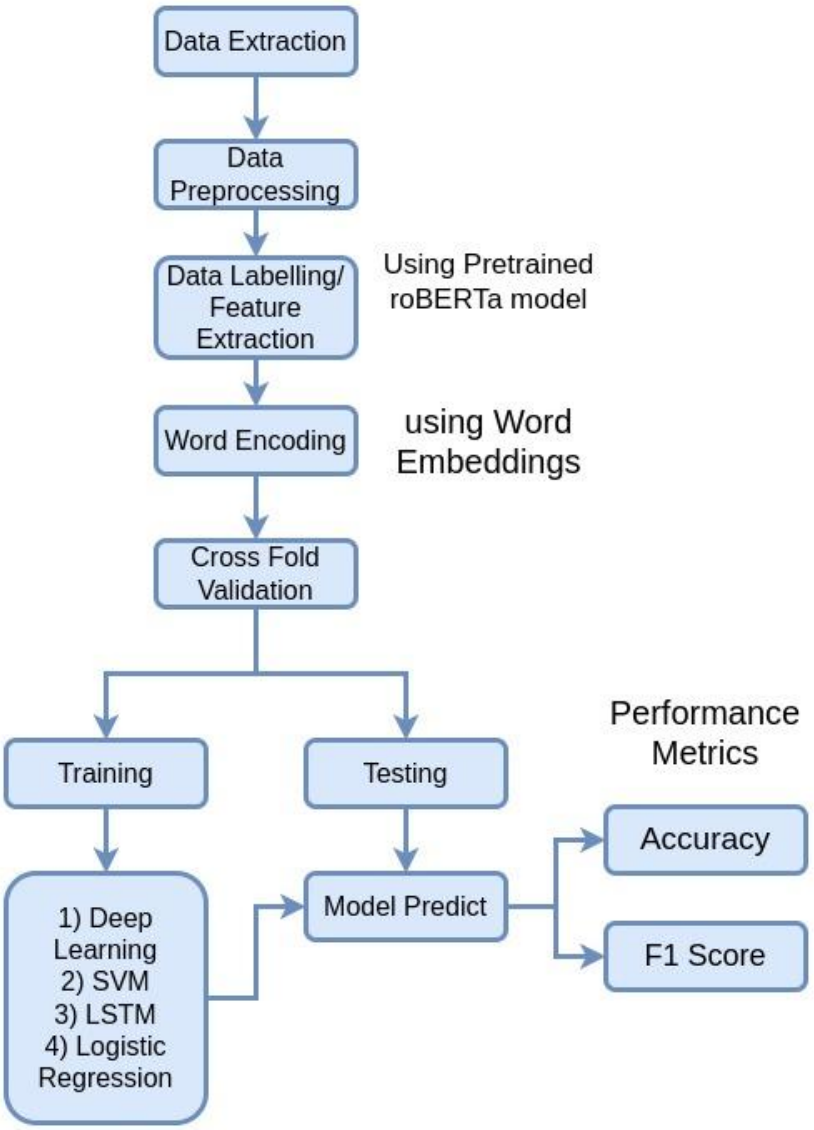


Fig 1

4.1 Dataset Extraction

After exploring a plethora of data sets ranging from labeled to unlabelled we have settled upon the **Movie Reviews dataset with around 25000** reviews along with their sentiments (i.e., Negative, neutral, positive), and **sentiment140 dataset** and **US Airline Sentiment dataset from Kaggle**, a trustable source of datasets.

4.2 Data Preprocessing

Once the dataset has been gathered then the process of Pre Processing starts, which was the padding of data, lemmatization and stemming of words, and transforming into a format that can be understood by the machine learning algorithm.

4.3 Word Encoding

Word encoding is a crucial step in sentiment analysis because it is necessary to convert the raw text data into a numerical representation that can be processed by machine learning algorithms. Sentiment analysis is the process of automatically identifying and extracting the sentiment of a given text, which can be positive, negative, or neutral. The sentiment of a text is determined by the words used and their context. Therefore, it is essential to represent words in a way that captures their meaning and relationships with other words.

Here in this project, we have used a technique known as word embedding.

4.3.1 Word Embeddings

Word embeddings are a more advanced technique that represents words as dense vectors of real numbers, where the values of the vector are learned during training. Word embeddings capture semantic relationships between words, which can improve the performance of the sentiment analysis model. Some popular

word embedding techniques include Word2Vec, GloVe, and FastText.

4.4 Data Labelling

Data labeling in sentiment analysis involves manually annotating a dataset with predefined labels that indicate the sentiment expressed in a text. It is essential because sentiment analysis relies heavily on the quality of the data it is fed, and inaccurate predictions can result in biased insights, misleading business decisions, and damage to brand reputation. Data labeling provides the ground truth for training machine learning models, which use the labeled data to learn patterns and make predictions on new, unlabeled data. Therefore, data labeling is critical to ensure the accuracy and effectiveness of sentiment analysis, and human annotators are still needed to ensure the accuracy and quality of the labeled data.

We have studied and tried labeling various well knows pre-trained models that are used to label unlabelled text.

Also, during the labeling study, we performed an accuracy check of pre-trained models using human-labeled data with the least error possible of TextBlob, Vader, and Roberta models.
And according to our study of pre-trained models given above **roBERTa model** should be the highest and our accuracy results even tell the same for it.

4.4.1 TextBlob Model:

TextBlob is a Python model for processing textual data, including sentiment analysis. It provides a simple API for labeling text data as positive, negative, or neutral based on a pre-trained machine-learning model. The TextBlob sentiment analyzer uses a **naive Bayes classifier** to identify the sentiment expressed in a text. It also supports advanced natural language processing features like noun phrase extraction, part-of-speech tagging, and language translation. The TextBlob model is easy to use and can be applied to various text analysis tasks, including social media monitoring, customer feedback analysis, and market research.

4.4.2 Vader Model:

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a **lexicon-based sentiment** analysis tool that uses a pre-built sentiment lexicon to assign a sentiment score to a text. It is a **rule-based model** that uses heuristics and linguistic rules to identify the sentiment expressed in a text. VADER not only identifies the sentiment polarity (positive, negative, or neutral), but it also considers the intensity of the sentiment. VADER is widely used in social media analysis and can handle informal language and slang.

4.4.3 roBERTa Model:

ROBERTA (Robustly Optimized BERT Pretraining Approach) is a language model built by Google for natural language processing (NLP) tasks, including sentiment analysis. It is based on the **BERT (Bidirectional Encoder Representations from Transformers)** architecture and uses a similar approach for training. However, ROBERTA employs additional training techniques, such as **dynamic masking and training on longer sequences**, to improve its performance. ROBERTA outperforms BERT and other NLP models in various tasks, including sentiment analysis. It is pre-trained on large-scale datasets and fine-tuned on specific tasks, making it highly

effective for various text classification tasks, including sentiment analysis.

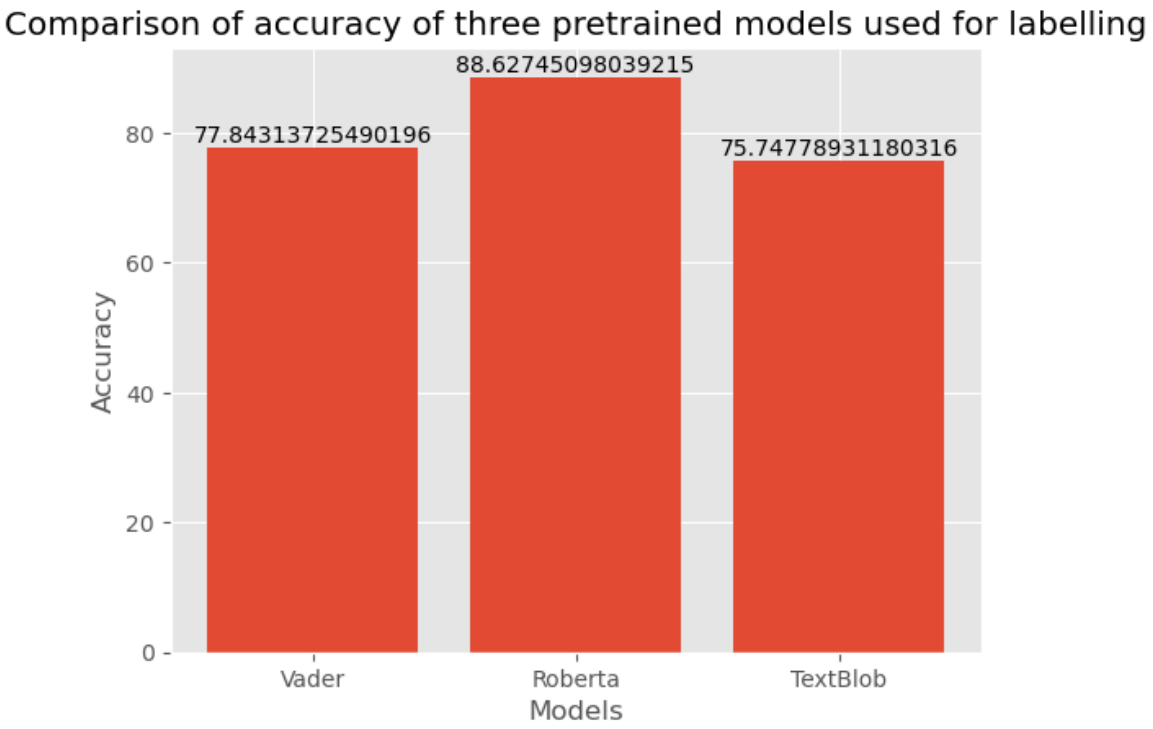


Fig 2

4.5 Training and Testing

As we had already tokenized and encoded the data in the word/data encoding step. We also have to set the limit on the length of each value of the dataset. The data was padded to a max of 5000 letters if greater than that, it was shortened down to 5000, and if greater, it was padded with zeroes.

Then the model was trained on the three following classification techniques, and the results with F1 score and accuracy were recorded.

a) **Deep Learning Model Architecture**

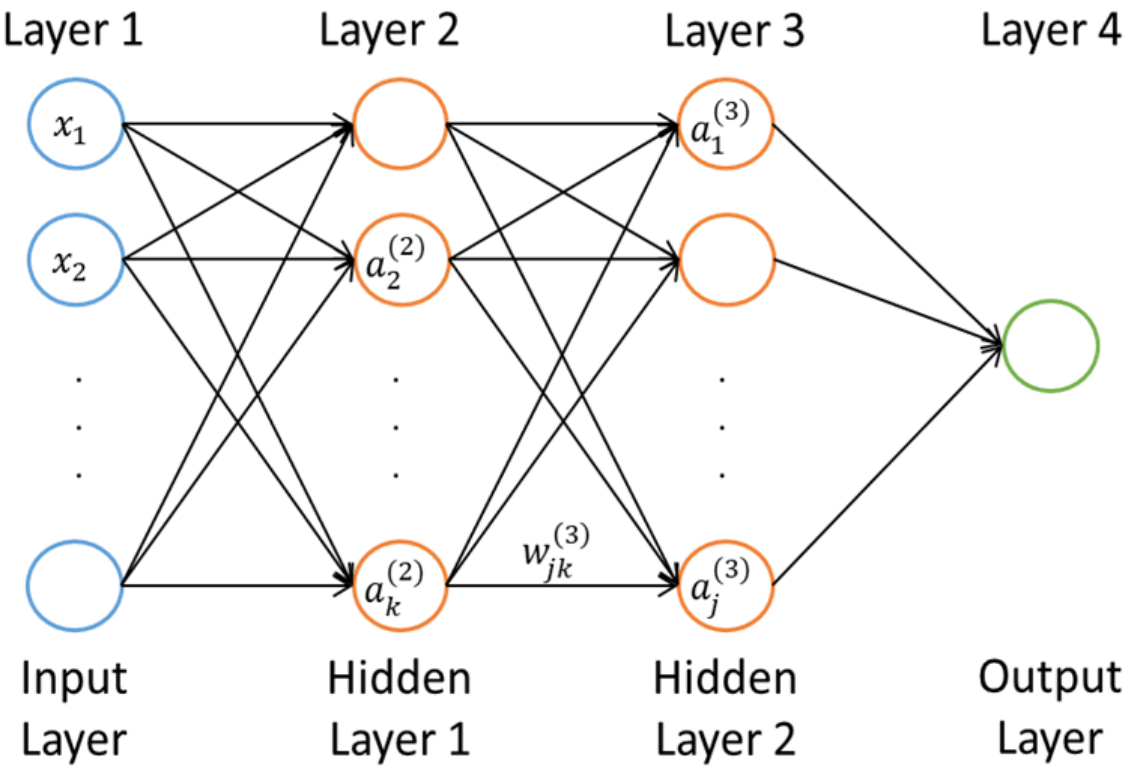
The Deep Learning Model used is of 4 layers, containing the following layers -:

1. **keras.layers.Embedding(10000, 16, input_length=500)** creates an embedding layer that converts integer-encoded text inputs (where each integer represents a word in a vocabulary of size 10,000) into dense vectors of size 16. The input_length parameter sets the length of the input sequences to 500. This layer learns an embedding matrix that maps each word to a dense vector representation in a lower-dimensional space.

2. **keras.layers.GlobalAveragePooling1D()** performs global average pooling over the sequence length axis of the input data. This layer takes the average of all feature maps across the entire sequence length, which helps to reduce the dimensionality of the input data and extract important features. The output of this layer is a 1D tensor of shape (16,).

3. **keras.layers.Dense(16, activation='relu')** creates a fully connected dense layer with 16 units and a ReLU activation function. This layer applies a linear transformation to the input data. It applies the ReLU activation function element-wise, which helps to introduce non-linearity to the model and extract more complex features.

4. **keras.layers.Dense(1, activation='sigmoid')** creates another fully connected dense layer with 1 unit and a sigmoid activation function. This layer produces a single scalar output representing the probability of the input text belonging to a positive class. The sigmoid activation



function maps the output to the range (0, 1), which can be interpreted as a probability.

b) SVM

Support Vector Machines (SVM) is a popular machine learning algorithm used for sentiment analysis. SVMs are binary classifiers that aim to find the hyperplane that best separates the positive and negative sentiment classes. The algorithm works by mapping the text data into a high-dimensional space and finding the optimal boundary that maximizes the margin between the two classes. SVMs are known for their ability to handle high-dimensional feature spaces and work well with small to medium-sized datasets.

c) Logistic Regression

Logistic Regression is another popular machine learning algorithm used for sentiment analysis. Unlike SVMs, Logistic Regression is a probabilistic algorithm that estimates the probability of a given input belonging to a particular class (i.e.,

positive or negative sentiment). The algorithm minimizes the error between the predicted probabilities and the actual labels. Logistic Regression is simple, interpretable, and works well with linearly separable datasets.

d) LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is commonly used for sentiment analysis. LSTMs can learn long-term dependencies in text data and can handle sequential input data. LSTMs are particularly useful for sentiment analysis as they can capture the context and meaning of words in a sentence and can learn from the order in which words appear in a sentence. LSTM models can be trained on large amounts of text data and handle variable-length inputs. They have achieved state-of-the-art performance on various natural language processing tasks, including sentiment analysis. LSTMs are widely used in marketing, social media analysis, and customer feedback analysis.

5. Results

For each feature dataset, a model was trained and tested on different types of models: **Deep Learning, Support Vector Machine (SVM), Logistic Regression (LR), and Long Short-Term Memory Networks (LSTM).** The **cross-fold validation F1-score** for each tuned model was performed and tabulated below.

Classifier	F1 Score	Accuracy
Decision Tree **[5]**	0.59	0.62
KNN **[5]**	0.57	0.60
GRU **[5]**	0.78	0.78
SVM	0.87	0.85
Logistic Regression	0.88	0.87
LSTM	0.86	0.83
(Proposed Model)Deep Learning	0.92	0.89

6. Featured Application

Some unique and practical applications which we have thought about sentiment analysis which we would be eager to explore in our future projects would be -

- **Customer Service Analysis**
  Speech to text (STT) conversion is also an major topic of study in machine learning. So using STT models, we will be converting the customer care call recording from speech to text, and then using sentiment analysis models it would tell about customer service analysis, which would eventually help a company better serve their consumers.

- **Healthcare:**
  Sentiment analysis can be used to monitor patient feedback and reviews of healthcare services and providers. This information can be used to improve patient experience and healthcare outcomes.

7. Future Aspects

As we move on to explore different techniques to improve performance, we found about hybrid modes. **Hybrid models** combine sentiment analysis techniques, such as lexicon-based, rule-based, and machine learning-based methods, to achieve more accurate and robust results.

Using a hybrid model in a major project has many benefits. For instance, it can help overcome the limitations of individual techniques and improve the accuracy and robustness of the sentiment analysis model. **Hybrid models can also handle various types of text data, including informal language and slang, by leveraging the strengths of each technique.**

Another promising future aspect of sentiment analysis is integrating **multimodal data, such as text, audio, and video, to provide a more comprehensive understanding of sentiment**. Combining different types of data can provide more insights into the user's sentiment and improve the sentiment analysis model's overall accuracy and effectiveness.

So there are major features which we would be exploring and working on in our major project works.

8. Conclusion

We concluded that for labeling, three models were studied (i.e., Vader, TextBlob and roBERTa) in which the transformers-based roBERTa model achieved the highest labelling accuracy.

In conclusion, this report has explored the application of sentiment analysis using four different models -**(Proposed Model)Deep Learning, Support Vector Machine (SVM), Logistic Regression (LR), and Long Short-Term Memory Networks (LSTM)**. The models were evaluated by labeled datasets to determine their performance metric (F1 score and Accuracy).
The results showed that the proposed Deep Learning model produced the highest **F1 score of 0.92**.

REFERENCES

1. **A comprehensive survey on sentiment analysis: Approaches, challenges and trends** by Marouane Birjali, Mohammed Kasri, Abderrahim Beni-Hssane LAROSERI Laboratory, Computer Science Department, University of Chouaib Doukkali, Faculty of Sciences, El Jadida, Morocco

2. **Sentiment analysis using product review data** by Xing Fang & JustinZhan (https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2)

3. **A Systematic Review of Predicting Elections Based on Social Media Data**: Research Challenges and Future Directions by Kellyton dos Santos Brito, Member, IEEE, Rogério Luiz Cardoso Silva Filho, and Paulo Jorge Leitão Adeodato.

4. **RoBERTa-LSTM: A Hybrid Model for Sentiment Analysis With Transformer and Recurrent Neural Network** by KIAN LONG TAN, CHIN POO LEE, KALAIARASI SONAI MUTHU ANBANANTHEN, AND KIAN MING LIM

5. Keras and Scikit API documentation (https://keras.io/api/) (https://scikit-learn.org/stable/modules/classes.html)