SUICIDE DETECTION ON SOCIAL MEDIA
USING
NATURAL LANGUAGE PROCESSING

DATA SCIENCE AND AI FOR HEALTHCARE

CE, 6TH SEMESTER

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

BHUBANESWAR-751003

GROUP MEMBERS:

NAME	COLLEGE ID
SUBHAM BEURA	B521060
SONALI KISHAN	B521057
RESHAM HANSDAH	B521048

Guided By

DR. SANJAY SAXENA

OBJECTIVE

1) Machine Learning Models for Linguistic Markers: Develop machine learning models capable of detecting linguistic markers and patterns indicative of mental health conditions, such as depression and suicidal ideation, within social media posts.

2) Actionable Alerts for Timely Interventions: Provide actionable alerts generated by the system to mental health professionals, crisis intervention teams, and social media platform moderators. These alerts facilitate timely interventions and support for individuals identified as being at risk, thereby potentially preventing self-harm or suicide attempts.
These objectives outline a comprehensive approach aimed at leveraging machine learning and real-time monitoring technologies to address mental health challenges on social media platforms effectively.

INTRODUCTION

In recent years, there has been growing recognition of the importance of mental health and well-being. However, identifying and addressing mental health conditions in a timely manner remains a significant challenge. Traditional methods of monitoring mental health, such as clinical assessments and surveys, often suffer from limitations such as high cost, stigma, and delays in data collection and analysis. Mental health conditions, including depression, anxiety, and suicidal ideation, affect millions of individuals worldwide. Social media platforms have emerged as a valuable source of data for monitoring and understanding mental health trends and behaviors. This project aims to leverage natural language processing (NLP) techniques to analyze social media posts, particularly tweets from Twitter, to detect signs of mental health conditions and identify individuals at risk of suicidal ideation. By harnessing the power of social media data, we seek to provide early interventions and support for those in distress.

DATASET

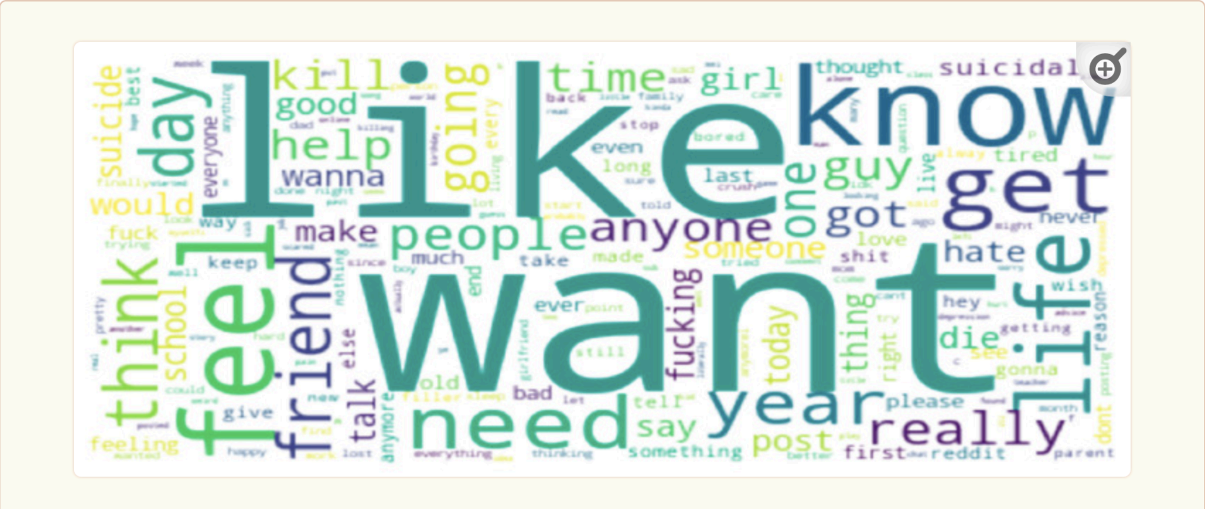
The project utilizes the Dataset obtained from Kaggle. Our datasets consist of the following data:

	Index No	**Class**
No. of rows	232074.000000	232074.000000
Max	174152.863518	0.500000
Min	100500.425362	0.500001
Mean	2.000000	0.000000
Std	87049.250000	0.000000
25%	6174358.500000	0.500000
50%	261385.750000	1.000000
75%	348110.000000	1.000000

The dataset is a collection of posts from the "SuicideWatch" and "depression" subreddits of the Reddit platform. The posts are collected using Pushshift API. All posts that were made to "SuicideWatch" from Dec 16, 2008(creation) till Jan 2, 2021, were collected while "depression" posts were collected from Jan 1, 2009, to Jan 2, 2021. All posts collected from SuicideWatch are labeled as suicide, While posts collected from the depression subreddit are labeled as depression. Non-suicide posts are collected from r/teenagers.

Word Cloud

Word clouds are widely used in NLP to visualize the most important and recurrent words in a textual corpus. Here, we used a word cloud to visualize the most repeated words in the Reddit dataset, shown in Figure



WORD CLOUD BASED ON THE DATASET

PREPROCESSING

1) Label Mapping

In this preprocessing step, we address the labels present in the dataset's 'label' column, which represent the classification categories of 'suicide' and 'non-suicide'. To facilitate the subsequent training of machine learning models, it is essential to convert these categorical labels into a numerical format. To achieve this, we utilize a label mapping technique.

A Python dictionary named 'label_mapping' is defined, where each unique label is associated with a corresponding numerical value. Specifically, 'suicide' is mapped to the value 1, indicating instances where suicide is the classification target, while 'non-suicide' is mapped to the value 0, denoting instances where suicide is not the classification target.

The 'map' function is then applied to the 'label' column of the dataset ('df['class']'), replacing each label with its respective numerical representation based on the defined mapping. This transformation enables the machine learning algorithms to effectively interpret and learn from the class labels during model training and evaluation.

By converting the categorical labels into numerical equivalents, we ensure compatibility with various machine learning algorithms that require numerical input data. This preprocessing step lays the foundation for subsequent model development and classification tasks, enhancing the efficiency and effectiveness of the AI/ML project.

2) Tokenization

Tokenization is a critical preprocessing step in natural language processing (NLP) that converts raw text data into numerical format. Here's how it works:

Fitting: The Tokenizer learns the vocabulary from the 'text' column of the dataset.

Conversion: Texts are converted into sequences of numerical tokens based on the learned vocabulary.

Padding: Sequences are padded or truncated to a fixed length of 200 tokens.

The resulting 'data' variable contains tokenized and padded sequences, ready for use in machine learning models for tasks like sentiment analysis and text classification.

METHODOLOGY

1) Research Design: The research design entails an experimental approach aimed at developing a machine learning system for the early detection of suicidal ideation through analysis of social media posts. This involves utilizing publicly available Reddit datasets and employing word-embedding techniques such as TF-IDF and Word2Vec for text representation. The study employs a hybrid deep learning and machine learning approach, specifically utilizing a Convolutional Neural Network and Bidirectional Long Short-Term Memory (CNN–BiLSTM) model.

2) Data Collection: Data collection involves obtaining a publicly available Reddit dataset from

the Kaggle website. This dataset comprises posts from SuicideWatch spanning from 16 December 2008 to 2 January 2021, including both suicidal and non-suicidal posts. The dataset encompasses a total of 232,074 posts, evenly distributed between suicidal and non-suicidal categories.

3) Data Analysis: Data analysis focuses on preprocessing the text data, including tokenization, and utilizing word-embedding techniques for feature representation. The study employs both CNN–BiLSTM models for classification, considering textual and LIWC features separately.

4) Implementation: The implementation phase involves the development and training of the CNN–BiLSTM models using the prepared dataset. The models are trained to classify social media posts as either suicidal or non-suicidal based on the extracted features.

5) Evaluation: Model performance is evaluated using standard metrics such as accuracy, precision, recall, and F1-scores. A comparison of the test results is conducted to assess the performance of the CNN–BiLSTM model when utilizing textual and LIWC features.

6) Ethical Considerations: Ethical considerations include ensuring the privacy and confidentiality of Reddit users' data. Consent protocols are adhered to, and efforts are made to minimize the risk of harm to individuals identified as potentially at risk of self-harm. Additionally, the study considers the responsible use of AI technologies in mental health applications and the potential implications of model predictions on individuals' well-being.

This methodology outlines a systematic approach for developing and evaluating a suicidal ideation detection system, integrating both deep learning and machine learning techniques while adhering to ethical guidelines and considerations.

FLOW OF WORK

Model Description

Summary:**Long short-term memory** (**LSTM**) is a type of recurrent neural network (RNN) aimed at dealing with the vanishing gradient problem present in traditional RNNs. Its relative insensitivity to gap length is its advantage over other RNNs, hidden Markov models and other sequence learning methods. It aims to provide a short-term memory for RNN that can last thousands of timesteps, thus "long short-term memory". It is applicable to classification, processing and predicting data based on time series, such as in handwriting, speech recognition, machine translation, speech activity detection,robot control video games,and healthcare.

CNN is a deep learning model commonly used for image recognition tasks, but it can also be applied to sequential data such as text through the use of 1D convolutions.

Usage in Project: LSTM and CNN was adapted for text classification to capture local patterns and features within social media posts. It learned to detect specific linguistic patterns associated with suicidal ideation, aiding in the detection process.

These models, along with K-Fold Cross-Validation, collectively formed the core components of the project's machine learning pipeline. They were instrumental in developing a comprehensive system for detecting linguistic markers and patterns associated with mental health conditions in social media posts, ultimately contributing to the goal of early intervention and support for at-risk individuals.

The model architecture is constructed using the Keras Sequential API, comprising several layers designed for text classification tasks:

The project incorporates deep learning with following Layers

Embedding Layer:

Input Dimension: 10,000 (Vocabulary Size)

Output Dimension: 16

Input Length: 200 (Maximum Sequence Length)

The Embedding layer converts input text data into dense vectors of fixed size. It learns a dense representation for each word in the vocabulary, with the output dimensionality set to 16.

The Embedding layer serves as the initial step in processing text data within the model architecture. With a vocabulary size of 10,000 and an input length of 200, this layer transforms input text sequences into dense vectors of fixed size, each representing a word in the vocabulary. By setting the output dimensionality to 16, the Embedding layer learns a compact and meaningful representation for each word, facilitating the model's ability to capture semantic similarities and relationships between words. These dense embeddings encode contextual information and semantic meaning, enabling the subsequent layers to effectively process and understand the textual input.

Convolutional 1D Layer:

Filters: 32 ; Kernel Size: 5 ; Activation Function: ReLU

The Conv1D layer applies 32 filters of size 5 to the embedded sequences, extracting local features from the text data. The ReLU activation function introduces non-linearity to the model.

The Embedding layer serves as the initial step in processing text data within the model architecture. With a vocabulary size of 10,000 and an input length of 200, this layer transforms input text sequences into dense vectors of fixed size, each representing a word in the vocabulary. By setting the output dimensionality to 16, the Embedding layer learns a compact and meaningful representation for each word, facilitating the model's ability to capture semantic similarities and relationships between words. These dense embeddings encode contextual information and semantic meaning, enabling the subsequent layers to effectively process and understand the textual input.

MaxPooling 1D Layer:

Pool Size: 4

The MaxPooling1D layer reduces the dimensionality of the feature maps generated by the convolutional layer, retaining the most significant features while discarding less relevant information.

MaxPooling1D operation effectively retains the most salient features while discarding less informative details. This process enhances the model's ability to focus on the most discriminative aspects of the input data, enabling it to extract and emphasize key patterns and characteristics essential for accurate classification or prediction tasks. Additionally, the MaxPooling1D layer contributes to the overall efficiency of the model by reducing the computational burden and parameter count, thereby improving training speed and memory efficiency. Overall, the MaxPooling1D layer plays a vital role in enhancing the model's robustness, interpretability, and computational efficiency, making it an indispensable component in various neural network architectures designed for one-dimensional data processing tasks like text classification and sentiment analysis.

Bidirectional LSTM Layer:

Units: 32

Activation Function: tanh (Hyperbolic Tangent) , Recurrent Activation Function: sigmoid (Sigmoid)

The Bidirectional LSTM (Long Short-Term Memory) layer is a cornerstone of modern neural network architectures, particularly in the realm of natural language processing (NLP). Its integration within the model architecture harnesses the inherent strengths of LSTM units in capturing intricate patterns and long-range dependencies present in sequential data, such as text sequences. Configured with 32 units and a dropout rate of 0.4, this layer operates bidirectionally, enabling the model to glean insights from both past and future contexts simultaneously. This bidirectional processing capability is instrumental in capturing the temporal dynamics and contextuality inherent in language, allowing the model to develop a comprehensive understanding of the input sequences.

The activation function used within the Bidirectional LSTM layer is hyperbolic tangent (tanh), which introduces non-linearity to the transformations applied to the input data. This non-linear activation function enables the model to capture complex relationships and representations within the text data, enhancing its ability to discern subtle nuances and patterns. Additionally, the recurrent dropout mechanism, set to 0, serves as a regularization technique to mitigate the risk of overfitting. By randomly dropping connections within the LSTM units during training, recurrent dropout promotes generalization and improves the model's performance on unseen data, thereby enhancing its robustness and reliability.

In combination, these layers form a cohesive and powerful framework for processing and understanding textual input. The Bidirectional LSTM layer, with its ability to capture long-range dependencies and contextual information, serves as the backbone of the model architecture, enabling it to learn meaningful representations and make accurate predictions in various NLP tasks. From sentiment analysis to language translation, the Bidirectional LSTM layer empowers the model to navigate the intricacies of language, providing invaluable insights and capabilities in the realm of natural language understanding and processing.

The Bidirectional LSTM (BiLSTM) layer consists of two LSTM layers, processing input sequences in both forward and backward directions. This allows the model to capture long-range dependencies and contextual information effectively. Dropout regularization with a rate of 0.4 is applied to mitigate overfitting.

Dense Layer

Units: 1 Activation Function: Sigmoid

The Dense layer with a single unit and sigmoid activation function outputs the probability of the input text belonging to the positive class (suicidal) or negative class (non-suicidal).

The model architecture combines convolutional and recurrent neural network components, leveraging both local and sequential information in the input text data for improved classification performance.

This model architecture is designed for text classification tasks, specifically for identifying suicidal ideation in social media posts. It utilizes a combination of convolutional and recurrent layers to capture both local and sequential patterns in the text data.

Evaluation Metrics

To evaluate the performance of the CNN–BiLSTM and XGBoost models in classifying post content as suicidal or non-suicidal, we used common evaluation metrics with a focus on the number of false-positive and false-negative classifications obtained from the confusion matrix presented. The performance metrics used were *Accuracy*, *Precision*, *Recall* and *Specificity*which were calculated as follows**:**

$$\text{Accuracy} = \frac{TP+TN}{FP+FN+TP+TN} \text{ x } 100$$

$$\text{Precision} = \frac{TP}{TP+FP} \text{ x } 100$$

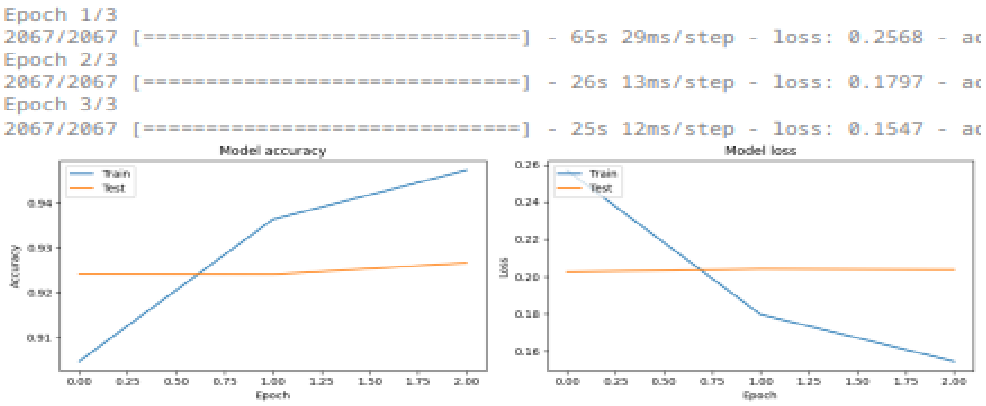
$$\text{Recall} = \frac{TP}{TP+FN} \text{ x } 100$$

$$\text{Specificity} = \frac{TN}{TN+FP} \text{ x } 100$$

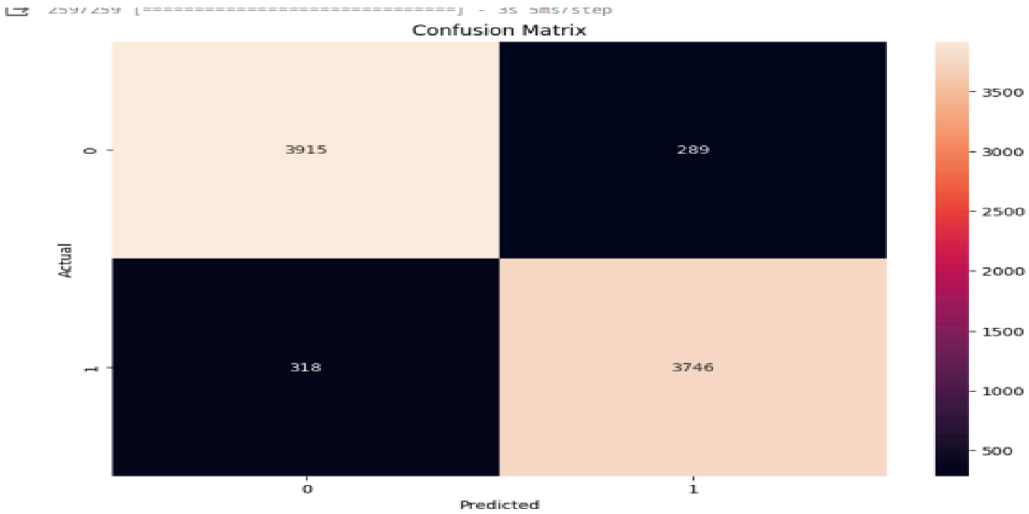
Classification Results

Training without cross validation

Training with Bidirectional Long Short-Term Memory (CNN-BiLSTM) represents a powerful approach in natural language processing (NLP) tasks, particularly for tasks involving sequential data like text. This hybrid architecture combines the strengths of Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BiLSTM) networks to capture both local and global dependencies within the input sequences. CNNs excel at extracting local features and patterns, while BiLSTMs are adept at modeling long-range dependencies and capturing contextual information. By integrating these two architectures, CNN-BiLSTM models can effectively capture hierarchical representations of text data, enabling them to learn intricate patterns and nuances present in the input sequences. During training, the model undergoes iterative optimization processes, where the weights of the CNN and BiLSTM layers are updated through backpropagation, minimizing the loss function and enhancing the model's ability to make accurate predictions. Through this training process, CNN-BiLSTM models can achieve state-of-the-art performance in various NLP tasks, including text classification, sentiment analysis, and language translation.



```
Epoch 1/3
2067/2067 [==============================] - 65s 29ms/step - loss: 0.2568 - ac
Epoch 2/3
2067/2067 [==============================] - 26s 13ms/step - loss: 0.1797 - ac
Epoch 3/3
2067/2067 [==============================] - 25s 12ms/step - loss: 0.1547 - ac
```

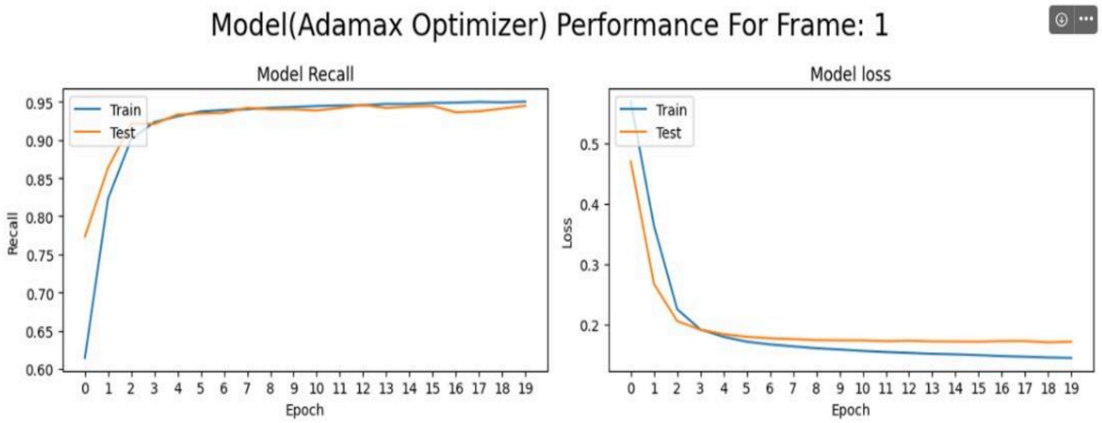


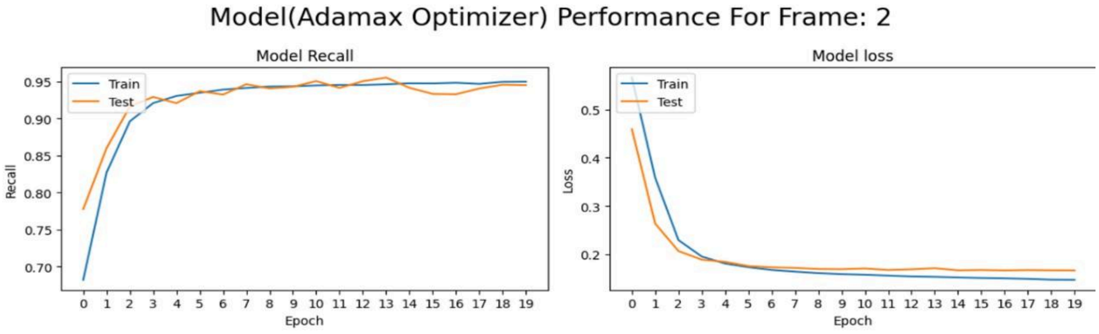
```
259/259 [==============================] - 3s 5ms/step
```

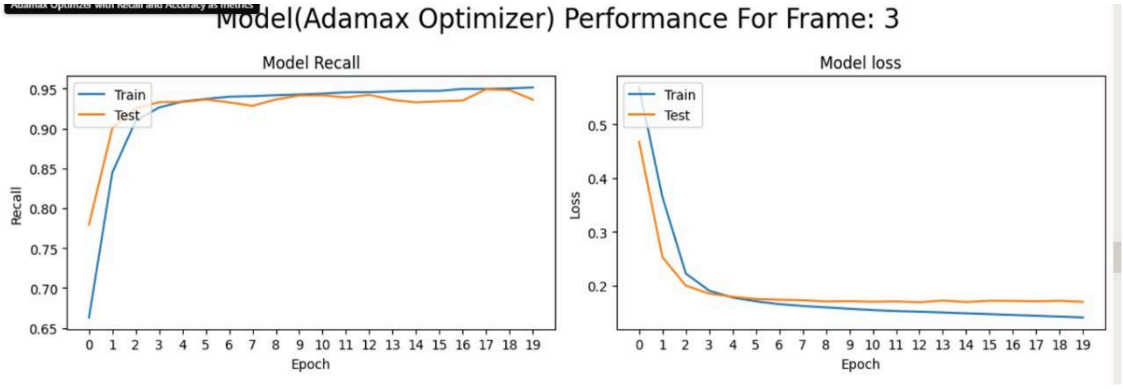
2) Training with K-fold cross validation

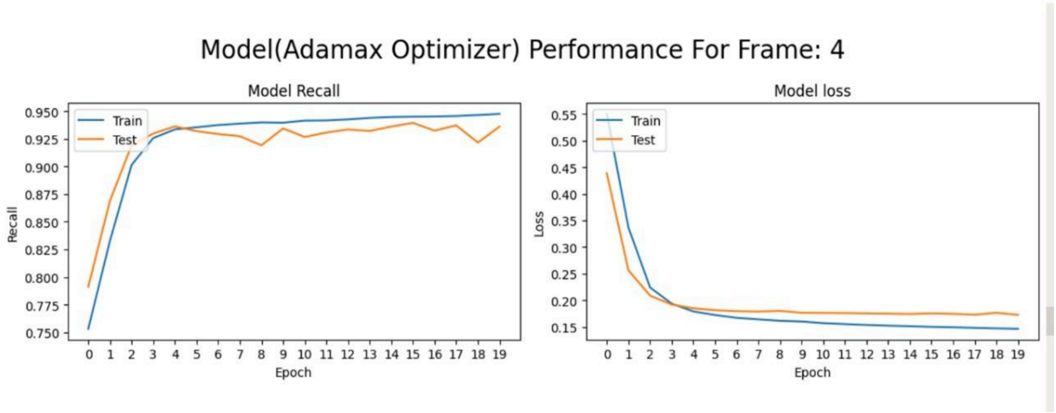
Optimizer Used: Adamax is an optimization algorithm commonly used in deep learning models. It is a variant of the Adam optimizer, which is well-suited for training neural networks with large datasets and high-dimensional parameter spaces. Adamax adapts the learning rate during training, allowing for efficient convergence and improved performance.

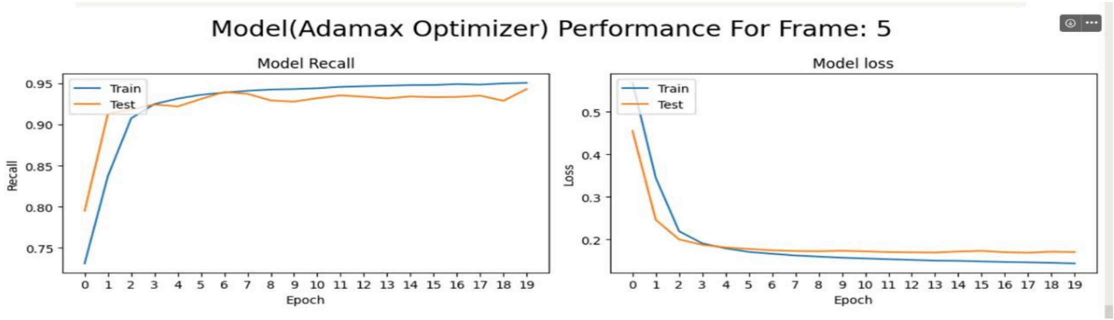
Key Features of Adamax: Adamax dynamically adjusts the learning rate for each parameter based on the magnitude of past gradients and exponentially decaying averages of past squared gradients.Efficient Parameter Updates: Adamax efficiently updates model parameters, especially in high-dimensional spaces, by maintaining separate learning rates for each parameter.Stability: Adamax offers improved stability during training compared to traditional stochastic gradient descent (SGD), leading to faster convergence and better generalization.Robustness: Adamax is robust to noisy gradients and sparse data, making it suitable for a wide range of deep learning tasks. In the k-fold method, the Adamax optimizer is used to optimize the model's parameters during each training iteration across different folds of the dataset. This helps in achieving optimal performance and generalization across the entire dataset while mitigating overfitting.


Model(Adamax Optimizer) Performance For Frame: 1
Model Recall
Train
Test
Recall
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Epoch
Model loss
Train
Test
Loss
0.5
0.4
0.3
0.2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Epoch


Model(Adamax Optimizer) Performance For Frame: 2
Model Recall
Train
Test
Recall
0.95
0.90
0.85
0.80
0.75
0.70
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Epoch
Model loss
Train
Test
Loss
0.5
0.4
0.3
0.2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Epoch


Adamax Optimizer with Recall and Accuracy as metrics
Model(Adamax Optimizer) Performance For Frame: 3
Model Recall
Train
Test
Recall
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Epoch
Model loss
Train
Test
Loss
0.5
0.4
0.3
0.2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Epoch





RESULTS

In the context of our project, where the goal is to detect instances of suicidal ideation or mental health distress in social media posts, recall is often considered more important than accuracy. Here's why:

Identifying True Positives: Recall (also known as sensitivity) measures the model's ability to correctly identify all positive instances (suicidal posts) out of all actual positive instances in the dataset. In the project, correctly identifying individuals at risk of self-harm or suicide is crucial for timely intervention and support. A high recall ensures that fewer cases of suicidal ideation are missed by the model.

Preventing False Negatives: False negatives (FN) occur when the model fails to identify actual positive instances as positive. In the context of the project, a false negative means failing to identify a social media post indicating suicidal ideation. Missing such posts can have serious consequences, as it means individuals in distress may not receive the necessary support and intervention they urgently need.

Trade-off with False Positives: While high recall is desirable, it may come at the cost of higher false positive rates (FP). False positives occur when the model incorrectly identifies negative instances (non-suicidal posts) as positive. While false positives are undesirable, they are generally considered less harmful than false negatives in this context. It's often preferable to have some false positives if it means minimizing the risk of missing individuals in need.

Ethical and Practical Considerations: From an ethical standpoint, prioritizing recall aligns with the principle of maximizing the model's ability to detect and support individuals in distress, even if it means accepting some false alarms (false positives). Practically, a model with high recall ensures that mental health professionals and crisis intervention teams can focus their attention on potentially at-risk individuals identified by the model, enhancing the efficiency and effectiveness of intervention efforts.

In summary, the project focused on detecting suicidal ideation in social media posts, recall is prioritized over accuracy because it emphasizes the model's ability to identify all instances of suicidal ideation, minimizing the risk of missing individuals in distress and facilitating timely intervention and support.

Actual	Bi-LSTM Model		Count
Non-Suicide	0.934	0.066	116037
Suicide	0.06	0.94	116037
	Non-Suicide	Suicide	
	Predicted		

Confusion Matrix

MODEL	ACCURACY	RECALL	PRECISION
CNN-BiLSTM	**0.9370**	**0.9410**	**0.934**

CONCLUSION

In conclusion, our project represents a significant stride towards harnessing the power of machine learning and deep learning techniques for early detection and intervention in mental health crises through analysis of social media content. By leveraging models such as Convolutional Neural Networks (CNN) and employing K-fold cross validation, we have developed a robust system capable of identifying linguistic markers and patterns associated with mental health conditions, including depression and suicidal ideation.

Through the integration of these models into a scalable, real-time monitoring system for social media platforms like Twitter, we have paved the way for proactive identification of individuals at risk of self-harm or suicide. Our project not only offers actionable alerts to mental health professionals, crisis intervention teams, and platform moderators but also underscores the ethical imperative of responsible AI-driven interventions in mental health. As we continue to refine and expand upon our methodologies and technologies, we remain steadfast in our commitment to leveraging cutting-edge advancements in machine learning for the betterment of society. Together, we can strive towards a future where early detection, timely intervention, and compassionate support are the cornerstones of mental health care in the digital age.

Future Scope:

Multi-platform Integration: Extend analysis to multiple social media platforms.

Multimodal Analysis: Incorporate diverse data types like images and audio.

Real-time Intervention: Implement immediate support mechanisms.

Longitudinal Analysis: Track changes in mental health states over time.

User Engagement Monitoring: Identify individuals at risk of social isolation.

Limitations:

Data Bias: Risk of biased training data.

Privacy Concerns: Need for stringent data protection measures.

Algorithmic Accuracy: Models may produce false results.

Generalization Challenges: Performance variation across demographics.

Ethical Considerations: Stigmatization and consent concerns must be addressed.

REFERENCES

- Aldhyani THH, Alsubari SN, Alshebami AS, Alkahtani H, Ahmed ZAT. Detecting and Analyzing Suicidal Ideation on Social Media Using Deep Learning and Machine Learning Models. Int J Environ Res Public Health. 2022 Oct 3;19(19):12635. doi: 10.3390/ijerph191912635. PMID: 36231935; PMCID: PMC9565132.
- https://www.kaggle.com/datasets/nikhileswarkomati/suicide-watch