# A Multi-Modal Analysis: Detecting Suicide Intention on Social Media

Jenifer Yu[1,†], Kelvin Mock[1,†] and Sabrina Cai[1,†]

[1]*University of Ottawa*

**Abstract**

Suicide has long been a significant global health concern. It causes harm to particular individuals. Health specialists are struggling to provide immediate help owing to a few reasons: partly because of the expensive and timely treatments required, and the shamefulness of individuals being diagnoses. We see the popularity of social media, where people tend to express their genuine thoughts. As a result, this project is formulated to perform a multi-modal analysis by mining some social media posts and eventually training some useful Natural Language Processing (NLP) based models which gives alarms to medical experts effectively whenever there is someone who is expressing negatively suicidal thoughts on the internet.

**Keywords**

Suicide Ideation, Data Privacy and Consent, Bias Mitigation, Web Scrapping, Linguistics, Sentiment Analysis, Multi-Modal Analysis, DORIS Framework, Transformers, Feature Extraction, Feature Contribution

## 1. Introduction

### 1.1. Problem Statement

Suicide is a significant global health concern, with millions of individuals expressing distress online before attempting self-harm. While social media platforms are supposed to provide a space where individuals share their thoughts and emotions, people who suffer from suicidal thoughts might find those platforms a valuable source to identify their suicidal intentions. However, this is never a healthy atmosphere since it encourages negative sentiments being spread over the internet and thus, yields a vicious cycle where people tend to believe expressing negative thoughts on the internet is an appropriate way. In turn, the detection of suicidal ideation on social media is a crucial task [1]. Therefore, this project intends to establish a multi-model solution in the detection of suicidal intentions through contents posted on social media platforms.

**Ethical Challenges** On the contrary, critics might pinpoint ethical concerns regarding privacy, consent, and data security [2]. Suicide detection using AI has gained attention as social media provides valuable real-time data for identifying individuals at risk [3]. Users may not be aware their data are being used, while there is a risk of misuse, making ethical oversight essential.

**An Ethical Solution** To mitigate ethical concerns, this project exclusively employs publicly-accessible datasets to ensure responsible AI practices. Most datasets were obtained from Kaggle [4] [5] [6] while one was obtained by scraping data from Reddit, abiding by the intended use and ethical terms [7] [8]. By enhancing predictive accuracy and maintaining ethical standards, this approach ensures user rights are protected while enhancing the predictive accuracy of the models that improve mental health monitoring.

---

[†]These authors contributed equally.

wyu094@uottawa.ca (J. Yu); kmock073@uOttawa.ca (K. Mock); hcai062@uottawa.ca (S. Cai)

**Linguistic Challenges** Technically speaking, detecting suicidal intentions through automated means poses another challenge due to the nuances of human language, sarcasm, slang, and varying contextual expressions, which are beyond the definition in an official dictionary. Therefore, as part of the modeling, this project aims to train an NLP-based model that is capable of accurately identifying suicidal intent in social media posts, helping early intervention efforts.

### 1.2. Objectives

The primary objective of this research is to build robust NLP models that detects suicidal intent effectively from social media text, which might also occasionally involve emojis. The system should efficiently analyze the sentiment of text, determine the likelihood of suicidal intent, and provide actionable insights that can assist mental health organizations in early intervention.

### 1.3. Research Gaps

Existing models for suicide detection might face limitations in the following criteria:

- Understanding contextual nuances and emojis within suicidal expressions;
- Handling imbalanced datasets where suicidal content is rare;
- Lack of real-time processing for early intervention; and,
- Ethical concerns surrounding data privacy and accuracy.

**Addressing the Gap** This project will address these challenges with advanced NLP techniques that we have learnt throughout the course, namely transformer-based models, as well as maintaining suitable ethical principles.

### 1.4. Research Questions

Hence, there are a few research questions:

1. How effective are transformer-based models (e.g., BERT [9], RoBERTa [10], DistilBERT [11] [12]) in detecting suicidal intent in social media posts compared to traditional machine learning models?
2. What linguistic patterns and sentiment markers are most indicative of suicidal intent in social media text?
3. Can hybrid models combining deep learning with rule-based filtering improve suicide detection performance?
4. How do different feature extraction techniques (e.g., sentiment analysis, emotion detection, word embeddings) impact the model's accuracy in identifying suicide intention?

## 2. Literature Review

### 2.1. DORIS Framework

**Why DORIS?** The World Health Organization reports 5% of adults suffering from depression. However, depression is not often diagnosed because of the expensive and timely medical process, and the shamefulness of individuals being diagnosed. The paper [13] sees the popularity of social media, where users tend to express their emotions genuinely online. As a matter of fact, the authors propose a detection framework - DORIS - for large-scale and low-cost monitoring.

**Challenges** Modeling an AI system for depression detection requires professional medical knowledge. Thus, it needs high accuracy and explainability for domain experts so that people will tend to rely on such a system. However, traditional machine learning classifiers lack medical interpretability, while LLM-based methods are explainable yet inaccurate. Furthermore, the authors pointed out that advent LLMs are sensitive to small prompt variations, leading to unpredictable non-deterministic results.

**Related Works** Early studies suggests using sentiment analysis and keyword detection. Some feature extraction techniques such as LIWC, TF-IDF, and LDA are also proposed. With those techniques, the paper prefers very simple classifiers, namely SVM or Logistic Regression, while it might also worth studying the difference of using deep-learning based transformers. The paper identified an improvement in explainability while using LLM-based methods. Hence, it is sensible to fine-tune an LLM (or a deep-learning based embedding model) to enhance accuracy.

**The Framework** **D**iagn**O**stic C**R**iteria-Guided Mood H**IS**tory-Aware, abbreviated as DORIS, is a framework which aims to enhance detection accuracy using the DSM-5 criteria, a widely-used scale aligning with psychological knowledge. It first takes the history of a user's post on social media, $P = \{P_1, P_2, \ldots, P_n\}$, with the corresponding timestamps, $t_1, t_2, \ldots, t_n$. This framework uses 3 types of features:

- Depression Symptom
- Post History Representation
- Mood Course Representation

At the end, to formulate a binary classification problem, it passes the scale as a prompt into an LLM and fine-tunes the annotated records with text embedding models. The scale is shown below:

1. Depressed mood
2. Loss of interest/pleasure
3. Weight loss or gain
4. Insomnia or hypersomnia
5. Psychomotor agitation or retardation
6. Fatigue
7. Inappropriate guilt
8. Decreased concentration
9. Thoughts of suicide

**Applying to this Project** We see the similarities of this scale to the suicide detection problem, essentially where it also indicates whether a particular post shows thoughts of suicide. As a result, we also pass this scale on to an LLM of our choice to annotate our data.

## 3. Datasets

In this project, we collected the necessary data (for modeling) from a combination of **web scraping** which adhered to ethical considerations [7] [8] as well as **publicly available datasets** in the domain of mental health and suicide risk detection. Since our models adhered to a supervised learning approach, we arranged those sources of data into a training set, a validation set, and a test set.

### 3.1. Training Set

A well-defined and labeled training set is crucial for a model to learn - or precisely, to update its weights according to the context.

**Twitter Suicidal Data** This dataset [5] contains 9199 posts on Twitter (informally called "tweets"), in a CSV (comma-separated valued) format. It contains 2 fields: tweets and the corresponding intention, which consists of binary labels with the underlying suicidal intentions. The advantage of this dataset is free of noise, because it does not bring any further irrelevant fields to our classification problem.

**Social Media Sentiments Analysis** This dataset [4] contains 732 posts from different social media platforms, also in a CSV format. It is served as a corpus marking the sentiment of each piece of text, which is perfect for subsequent sentiment analysis. It contains 15 fields in total, which provides many insights of a particular post such as the platform where it has been posted, the associated hashtags, the number of likes, the country where the user is located in, and more. The abundance of attributes provide a good context of analyzing the ethical fairness and bias across different identifiable subgroups. Moreover, posts are labeled by prompting to the state=of-the-art generative Large Language Model (LLM) - Google Gemini. They are treated as the "true label" during the classification task.

### 3.2. Validation Set

**Reddit SuicideWatch Posts** This dataset was created through web scrapping. The maximum number of posts allowed to be scrapped at once is 100. By launching a web request to Reddit ethically [8] [7], we load a list of 100 records of posts from "https://reddit.com/r/SuicideWatch/new.json?limit=100", which is in the "subreddit" (i.e., category) named "SuicideWatch". Raw data was then loaded into a JSON file. We annotate all posts here to be suicidal, leading to potential class imbalance in the validation set. Therefore, resampling is needed prior to modeling.

### 3.3. Test set

**Depression Tweets** This dataset [6] contains 18679 tweets. It has only 1 field, which is the content of the posts, in a JSON format. It is also free of noise in our classification problem. Since records here are not labeled, which means that ground truths are not provided, those data are treated as a "test set", mainly demonstrating the prediction ability and strength of models in this project.

## 4. Methodologies

A description of the method you have designed or of the methods you are comparing. Assume that the reader does not know how the systems you have designed and/or used work.

### 4.1. Data Pre-Processing

**Steps** To pre-process the raw datasets, we performed the following steps in sequence:

1. **Data Cleansing**: removing single-valued columns and replacing invalid NaN values.
2. **Emojis Normalization** [14]
3. **Symbols Normalization** [15]: including `- @ # http`
4. **Punctuations Normalization** [15]
5. **Lowercase Conversion**
6. **Lemmatization**: converting abstract forms of words to their base forms, for example, "running" to "run".
7. **Word Tokenization**: relying on this Regular Expression (ReGex) - `\b\w+(?:'\w+)?\b`
8. **Stopwords Normalization** [16]
9. **Text Vectorization**: converting texts to numeric matricies using DistilBERTTokenizer [12] which preserves contextual meanings well.

**Extracting Sensitive Attributes** We also extracted sensitive attributes for ethical bias analysis across subgroups (especially in the "Social Media Sentiments Analysis" dataset). We prompted Google Gemini to identify sensitive attributes from the datasets.

**Annotation** Some datasets do not show an accurate label identifying whether the corresponding text is suicidal. Like the "Social Media Sentiments Analysis" dataset initially did not exhibit a field with labels for our problem. To identify the true labels, Google Gemini is used for annotation, which was prompted according to the psychological DORIS scale [13], with 0 to 9 labels. From the framework, the label 0 is added to the scale to indicate that a post is positive or neutral.
To align with the problem of classifying whether a post exhibits a suicidal intention, labels 0 to 8 are interpreted as non-suicidal, which will be converted to 0; whereas label 9 is interpreted as suicidal, which will be converted to 1.

## 4.2. Models

The datasets contain user-generated text content, allowing NLP models to assess different levels of suicidal risk. We will make a comparison across multiple NLP models.

### 4.2.1. Baseline Model

This model takes text embeddings in a shape of (3, 768), which represent the vectorized segments of texts - title, content, and hashtags. Due to class imbalance, SMOTENN is applied. It is a resampling technique that combines undersampling and oversampling. Afterwards, a collection of simple machine learning models are defined for a comparison, with mostly default hyperparameters.

**A Collection of Models** We have a K-Nearest-Neighbor (KNN) classifier, a Logistic Regression classifier, a Support Vector Machine (SVM) classifier, a Decision Tree classifier, and a Naive Bayes classifier. Those models predict in a binary classification problem with a probabilistic output. By comparing these models with their macro-F1 scores, we concluded that the Logistic Regression classifier is the best model. See figure 1 in the "Results" section. Hence, it is identified as the baseline model in this project.

### 4.2.2. Deep Learning based Model

This model is formed by a pipeline of 2 parts: **fine-tuning a pre-trained DistilBERT** transformer [11] [12], followed by **training some custom layers**. It takes the raw text as an input into the fine-tuned DistilBERT transformer, and gives a probabilistic output from those custom layers. Due to class imbalance, data are randomly stratified into training and validation set upon data collection.

**Fine-tuning** What defines "fine-tuning" is the process of training the pre-trained "distilbert-base-uncased" model again with the data in the context of suicidal detection. It adjusts the weights from the transformer such that it fits better into context. However, the challenge stems from varying lengths of posts. As a result, texts are truncated to a maximum of 128 characters, and the remaining parts are padded with empty spaces known as "attention masks". By fine-tuning it, it goes through a cross-validation process, which further trains the model with the best set of hyperparameters. A model in this phase of fine-tuning is trained with 5 epochs. Afterwards, part of the field embeddings are extracted to be further trained in some custom layers. To be specific, 32 characters are extracted from the text embeddings. Before stepping into the next phase, SMOTENN is also applied to both the training and validation data. It efficiently copes with unbalanced datasets.

**Custom Layers** With the partial embeddings being extracted, a Tensorflow-based Keras model is built with several custom layers apart from the ones in the DistilBERT transformer. It takes the array of embeddings as an input. A **bidirectional LSTM (Long-Short Term Memory) layer** captures temporal relationships and contextual dependencies from 3 segments of the texts - title, content, and hashtags. Suicide-related cues might arise from interactions of a post's title, content, and hashtags. The LSTM layer reads the sequence forward and backward, enhancing the model's understanding of such dependencies. A **dropout layer** effectively prevents overfitting by randomly deactivating a fraction of neurons during training. Suicide detection datasets are often imbalanced and small. Dropout thus adds regularization to reduce the generalization error. A **dense layer** learns non-linear combinations of features extracted by the LSTM. Stacking dense layers allows the model to learn hierarchical abstractions, improving the decision boundary for the binary classification between suicidal and non-suicidal posts. Finally, the model outputs a probability between 0 and 1 for the classification. Overall, the custom layers can be seen as a lightweight LSTM-based classifier, placed on top of fixed DistilBERT-based embeddings. After obtaining the best set of hyperparameters, the resulting best model is then trained with 20 epochs, with 5 folds of cross-validation. These custom layers focus on: extracting temporal signals between different text segments, enhancing decision making with deep feedforward layers, and maintaining generalizability through dropout and cross-validation. Those custom layers are experimented in a standalone setting, but showing 1.0 in accuracy, precision, recall and F1 scores seems to be overfitting. Hence, those custom layers must be added on top of the fine-tuned DistilBERT model which handles generalization much better.

**Hyperparameters** Hyperparameters are tuned with a cross-validation process. The DistilBERT model is tuned within 3 folds, in which each fold takes turns to be the validation set. In each fold, data are divided into training and validation sets. The training set is the major part of the tuning, and the validation set is to evaluate how well a model generalizes to unseen data and thus to prevent overfitting. Moreover, those custom layers also go through cross-validation and a manual hyperparameter tuning process. It is an exhaustive grid search with potential LSTM units, dropout rates, dense units, and learning rates for the layers. The tuning trains a version of the classifier (with custom layers) within 5 epochs. Such a process is significant because it ensures the model will not predict overly accurate by memorizing the training data. As a result, such a structure copes well with generalization to reduce the risk of overfitting.

**Best hyperparameters across all folds** `{'lstm_units': 64, 'dropout_rate': 0.1, 'dense_units': 32, 'learning_rate': 1e-05}` with validation accuracy: 0.7069.

#### 4.2.3. LLM-based Models

This model relies on making prompts to a Deepseek-based [17] transformer using PyTorch.

**Model's Input** This model uses AutoTokenizer to tokenize raw texts into vector embeddings. The maximum possible length of text of a post is defined to be 512 characters, where the remaining parts will also be padded with attention masks.

**Fine-tuning the Model** Like training DistilBERT, the pre-trained "deepseek-ai/deepseek-llm-7b-base" is also fine-tuned, but within 3 epochs. The batch size for both the training and validation sets is defined to be 4. It takes 100 steps to warm-up. Warmup steps refers to the proportion of training to be dedicated to a linear warmup where learning rate gradually increases [18]. The weight decay rate is applied for regularization. It is defined to be 0.01. It helps in preventing the model from overfitting by penalizing large weights [18]. Eventually, this model also gives a probabilistic prediction as an output.

#### 4.2.4. Hybrid Model

A hybrid model uses an ensemble method to combine two models above, which are the baseline model and the DistilBERT-based model. Please check the results in the next section. The hybrid model is a customized voting classifier which relies on a combination of voting strategies. A soft voting decides the label for a certain piece of text based on an array of predicted probabilities from each class. It outputs the class with a higher probability. A hard voting is a label-based strategy that simply performs majority voting. When both strategies give a predicted label, it decides the resulting prediction with an "AND" operation. In other words, if both methods predict non-suicidal, the resulting prediction is non-suicidal, or otherwise, it is predicted suicidal.

### 4.3. Evaluation Methodologies

We evaluated the performance of a model using a standardized framework of metrics: accuracy, precision, recall, macro-F1, micro-F1 and Area Under the Curve (AUC). It is measured in the validation set since the test set is unlabeled. After obtaining the metrics, we show their predictions based on the unlabeled test set. We also analyze the meaning of words with a strucured sentiment analysis.

#### 4.3.1. Metrics

**Accuracy** The accuracy is calculated simply on the basis of the ratio of correctly classified samples.

**Model's Output Quality** The quality of a model is measured on the basis of true positive labels, false positive labels, true negative labels and false negative labels. Please note that positive labels refer to being identified as "suicidal" in this context. $T_p$ represents the number of correctly classified positive labels (i.e., true positives) and $F_p$ represents the number of falsely classified positive labels (i.e., false positives). True negatives $T_n$ and false negatives $F_n$ are represented similarly. Those metrics could be visualized in a confusion matrix. In our problem, our goal is to maximize the true positive rate and to prioritize minimizing the false negative rate, because a medical diagnosis is preferred to cover as much potentially suicidal patients as possible.

**Precision** The precision, also known as the "positive predictive value", is calculated with this formula: $P = \frac{T_p}{T_p+F_p}$. It measures the ratio of true positives from the classification of all positive predicted labels.

**Recall** The recall, also known as the sensitivity, is calculated with this formula: $R = \frac{T_p}{T_p+F_n}$. It measures the ratio of true positives from the positive labels based on the ground truths.

**F1 Score** Since the precision and recall tells different conclusions, we balance the trade-offs with the F1 score. The F1 score is calculated with this formula: $F_1 = \frac{2 \cdot P \cdot R}{P+R}$. There are 2 types of a F1 score. Macro-F1 calculates the unweighted mean of the F1 scores for each class, while micro-F1 aggregates the true positives, false positives, and false negatives across all classes and then calculates the F1 score [19].

**Area Under the Curve** AUC measures the change between the true positive rates and the false positive rates, which is another way to balance the biased conclusions drawn from the precision and recall. It can be visualized on a Receiver Operating Characteristic (ROC) curve.

**Word Importance** Word importance is an important metric in NLP showing which words are highly contributing to the predictions within the corpus. Tokenized strings, which are individual words in this context, are features in the predictions. Measuring feature contribution/importance is a common approach to analyze how the model processes the corpus of words, and even the biases of a model.

#### 4.3.2. Sentiment Analysis

Regarding the research question about what linguistic patterns and sentiment markers are the most indicative of suicidal intent in social media text, a **sentiment analysis** has been performed by firstly extracting the top 20 **n-grams** and afterwards visualizing the **polarity and emotional strengths** of words in those n-gram patterns with TextBlob [20] and VADER [21].

**N-Gram** N-gram represents a collection of $n$ consecutive words, where $n$ is a predefined constant. We investigated word patterns from 2-gram to 5-gram. It is a common technique to understand what word patterns appearing often in the datasets.

**Polarity** Based on the common N-grams, the polarity score is a float within the range [-1.0, 1.0] inclusively. A score of -1 means the words are super negative, like "disgusting" or "awful." A score of 1 means the words are super positive, like "excellent" or "best". It is helpful to analysis what kinds of patterns are most likely indicated suicidal, like whether they are tend to me positive or negative, and to what extend of it. In this analysis, polarity is measured with TextBlob [20] and VADER [21].

**Emotion Distributions** The average distribution of emotions tells us how a dataset is biased in terms of the type of words. For example, from a radar chart, we can easily compare the ratios of words with positive, negative and neutral meanings respectively.

**Model Sensitivity** Although the recall tells us the sensitivity from a certain array of predictions, the model's sensitivity here tell us how the number of negative sentiments or the likelihood of predicting a text to be suicidal affects predictions. In other words, it tells how likely a model predicts suicidal.

**Subjectivity** The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective. It is calculated using TextBlob [20]. It tells normally how subjective texts from each predicted class are. It helps identifying the sentiment in general.

## 5. Results

### 5.1. Baseline Model Comparison

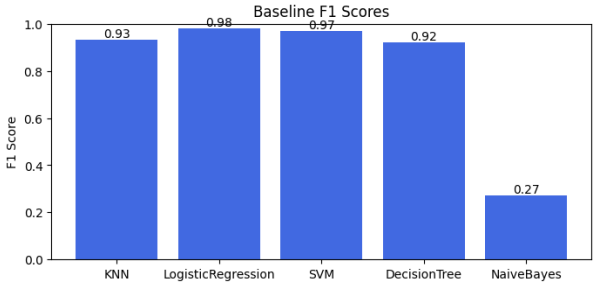



Figure 1: A Comparison of Baseline Models based on Macro-F1 scores

**Defining our Baseline** This plot shows that the Logistic Regression classifier shows the highest macro-F1 score among others, and therefore, it is used as the baseline model in the analysis.

## 5.2. Results of the Baseline Model

### 5.2.1. Model's Performance

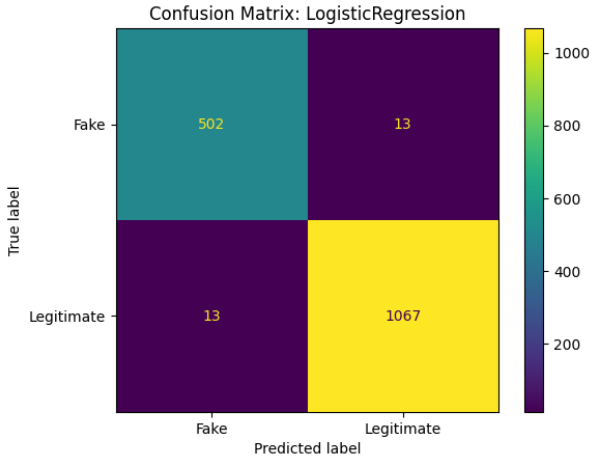



Figure 2: Confusion Matrix of the Logistic Regression classifier as the Baseline

**Prediction Metrics** From the confusion matrix, it seems that the baseline model successfully keeps the false positive and false negative rates minimal. It is important since we want to correctly classify suicidal posts for intermediate diagnosis. As a result, we conclude the results with the following metrics: **Accuracy**: 0.9837; **Precision**: 0.9814; **Recall**: 0.9814; **Macro F1**: 0.9814; **Micro F1**: 0.9837; **AUC (area-under-the-curve)**: 0.9981.

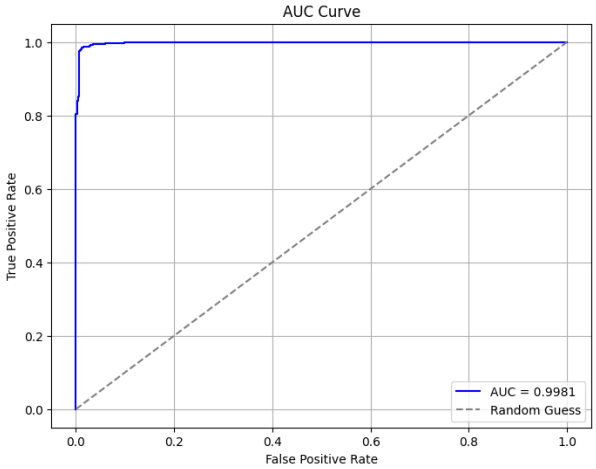



Figure 3: ROC-AUC Curve of the Logistic Regression classifier as the Baseline

**Finding Insights from the AUC** The ROC curve shows a very high AUC, of 99.81%. The good side is the high accuracy. But the trade-off might be the risk of overfitting. Therefore, we need a comparison with some advanced models like the use of transformers in our multi-modal analysis.

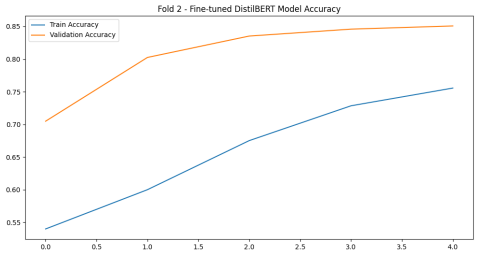
## 5.3. Results of the Deep Learning based Model

This model took almost 3 hours to fine-tune a DistilBERT transformer, and more than 30 hours to train custom layers with 20 epochs. A more detailed summary of execution times is shown below:

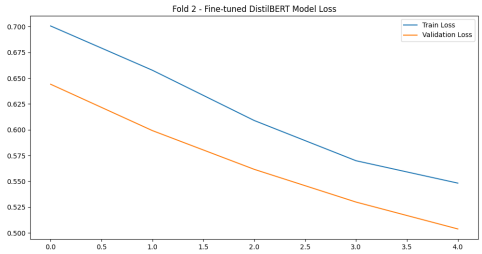
- Step 1 (data extraction and preprocessing): 366.6863784790039 seconds (6.11 hours)
- Step 2 (randomization and split): 0.011476278305053711 seconds

- Step 3 (fine-tuning DistilBERT): 8236.603701353073 seconds (2.29 hours)
- Step 4 (embedding extraction): 1975.3452780246735 seconds (32.92 minutes)
- Step 5 (SMOTENN): 5.4160315990448 seconds
- Step 6 (dataset preparation): 0.713801383972168 seconds
- Step 7a (tuning the best custom classifier model within 5 epochs): 0.11855673789978027 seconds
- Step 7b (building the custom classifier model within 20 epochs): 0.1316518783569336 seconds
- Step 8 (cross-validation and hyperparameter tuning): 112068.81693029404 seconds (31.13 hours)

### 5.3.1. Model's Performance

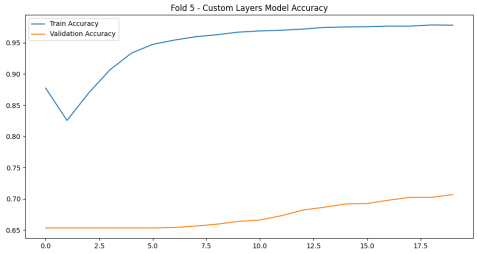



(a) Accuracy of Fine-tuning DistilBERT (Fold 5)

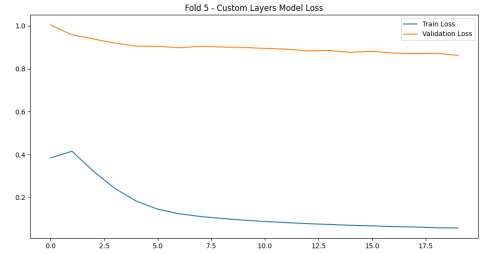



(b) Loss of Fine-tuning DistilBERT (Fold 5)

**Training Curves** By fine-tuning a pre-trained DistilBERT model, fold 5 seems to be the best fold. However, the accuracy using the training data still seems much lower than using the validation data. Similarly, the loss using the training data seems much higher than using the validation data. This is a sign of underfitting. This might stem from inefficient number of training epochs. The fine-tuning process is early stopped before convergence, causing the accuracy not to reach a specific standard.




(a) Accuracy of Custom Layers (Fold 5)




(b) Loss of Custom Layers (Fold 5)

**Training Curves** With custom layers, the accuracy of using the training data is now much higher than using the validation data. Similarly, the loss of using the training data is much lower than using the validation data. For one thing, this model is trained on more number of epochs with those custom layers. For another, it shows convergence, which is a sign of stablizing the model's performance. However, with a high accuracy, the risk of overfitting might occur.

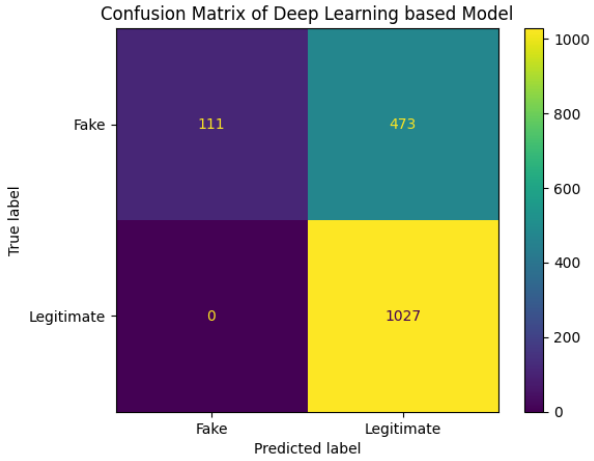



Figure 6: Confusion Matrix of the DistilBERT-based Deep Learning Model

**Prediction Metrics** From the confusion matrix, it seems that the DistilBERT-based model shows 0 false negative rate. It leads to the recall being 1.0, which is a sign of overfitting. This tells us that, among those texts that are truly suicidal, this model will 100% indicate it suicidal correctly, while on the other hand, when a text is non-suicidal, it fails to keep the false negative rate minimal. More significantly, this model indicates all suicidal cases correctly for prompt medical diagnosis, although some of them might be falsely indicated positive. As a result, we conclude the results with the following metrics: **Accuracy**: 0.7064; **Precision**: 0.6847; **Recall**: 1.0000; **Macro-F1**: 0.5661; **Micro-F1**: 0.7064; **AUC (area-under-the-curve)**: 0.5950.

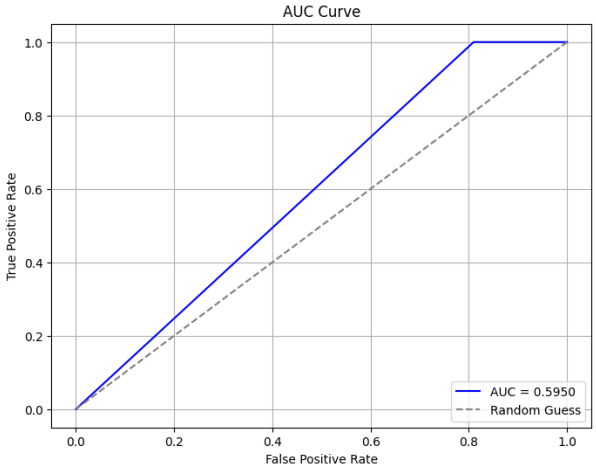



Figure 7: ROC-AUC Curve of of the DistilBERT-based Deep Learning Model

**Finding Insights from the AUC** The ROC curve shows a moderate AUC, of 59.50%. It is close to the outcome of taking a random guess with 50% accuracy. As discussed, it might be because it overly indicates suicidal texts. With a more reliable model in a production-grade environment, the use of a sophisticated LLM or a model with a combined strategy may be more desirable.

## 5.4. Results of the LLM-based Model

### 5.4.1. Model's Performance

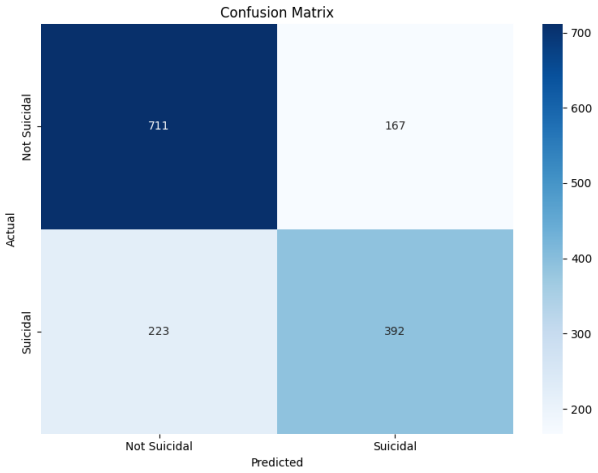



Figure 8: Confusion Matrix of the LLM-based Model

**Prediction Metrics** From the confusion matrix, it seems that the LLM-based model also successfully keeps the false positive and false negative rates minimal. It is ensures correct classification during an intermediate diagnosis. As a result, we conclude the results with the following metrics:
**Accuracy**: 0.7388; **Precision**: 0.7013; **Recall**: 0.6374; **Macro-F1**: 0.6678; **AUC (area-under-the-curve)**: 0.7236.

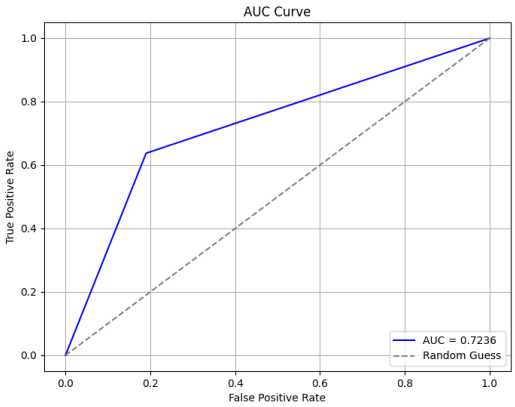



Figure 9: ROC-AUC Curve of the LLM-based Model

**Finding Insights from the AUC** The ROC curve shows a moderate high AUC, of 72.36%. It is slightly enhanced from the DistilBERT-based model. This proves how sophisticated an LLM is in terms of natural language predictions.

## 5.5. Results of a Hybrid-based Model

The hybrid-based model combines 2 models. The baseline model in use relies on text embeddings extracted from the raw pretrained DistilBERT tokenizer. The deep learning based model in use of this hybrid-based model relies on text embeddings extracted from the fine-tuned DistilBERT model. Therefore, there are 2 different sets of predictions, visualized in 2 confusion matricies below, respectively.

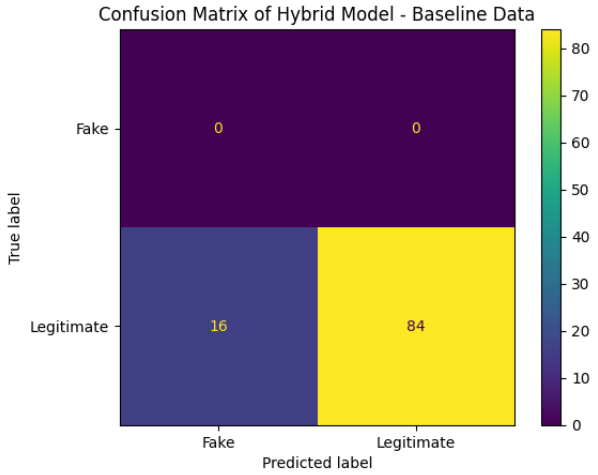



Figure 10: Confusion Matrix of the Hybrid-based Model with Baseline Data

**Prediction Metrics** From the confusion matrix, it seems that the data from the baseline model shows class imbalance, where no non-suicidal texts are in the dataset. It will be difficult to robustly show the model's performance, such as with a ROC-AUC curve. Anyways, we conclude the results with these metrics:
**Accuracy**: 0.8400; **Precision**: 1.0000; **Recall**: 0.8400; **Macro-F1**: 0.4565; **Micro-F1**: 0.8400; **AUC (area-under-the-curve)**: 0.0000.

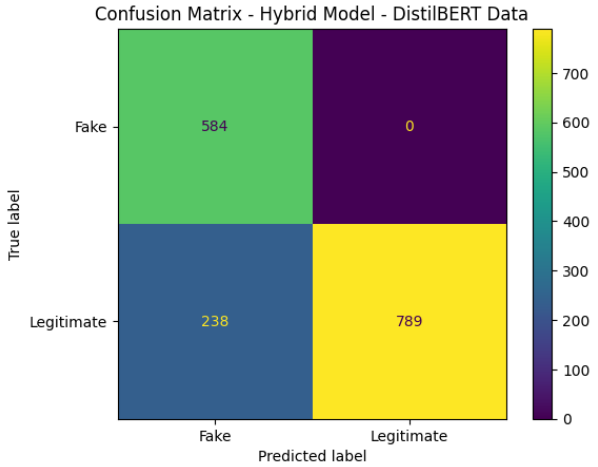



Figure 11: Confusion Matrix of the Hybrid-based Model with Fine-Tuned DistilBERT Data

**Prediction Metrics** From the confusion matrix, it seems that the data from the fine-tuned DistilBERT model now shows a well balanced dataset, probably because the pipeline in the second model goes through SMOTENN to resample the data. Now we can visualize it even with a ROC-AUC curve. Therefore, we conclude the results with these metrics:
**Accuracy**: 0.8523; **Precision**: 0.8950; **Recall**: 0.8523; **Macro-F1**: 0.8498; **Micro-F1**: 0.8523; **AUC (area-under-the-curve)**: 0.8841.

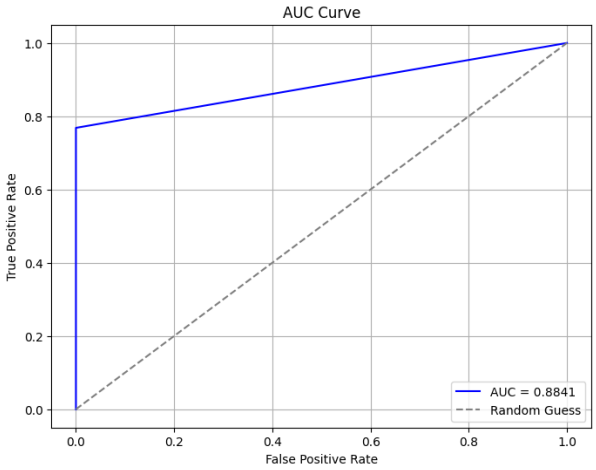



Figure 12: ROC-AUC Curve of of the Hybrid-based Model

**Finding Insights from the AUC** The ROC curve shows a high AUC, of 88.41%. It improves from the DistilBERT-based model but not as high as the baseline model's. This shows the capability of the model of balancing the accuracy and the risk of overfitting, ensuring a more robust and reliable model in use.

## 5.6. Word Importance

Furthermore, we summarized some importance words from the corpus in 5 samples respectively. Please refer to graphs in the appendix section.

**Bag of Words** An analysis has summarized some important words. For example, the text "m0ni" has attained an importance score of almost 1.0 in the third sample. The word "just" has attained an importance score of almost 1.0 in the fourth sample. It inferences the Deepseek model without further training, in order to deduce the features' contributions, which refers to which words contribute highly to suicidal risk predictions.

## 5.7. Sentiment Analysis

### 5.7.1. Polarity

The graph below shows the mean polarity from top 20 n-grams across different datasets. You may also find polarity scores evaluated by each method in the appendix section.

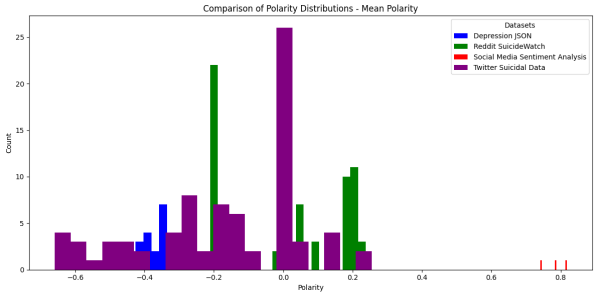



Figure 13: Distributions of Mean Polarity of Word Patterns Across Datasets

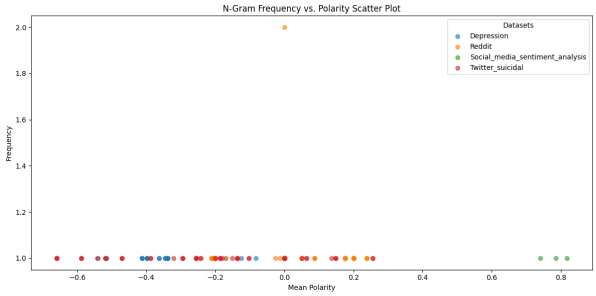



Figure 14: Distributions of Mean Polarity of Word Patterns from **Different N-Grams** Across Datasets

**Polarity VS Model's Prediction** The following heatmaps show comparisons of mean polarity scores versus a certain model's predictions.

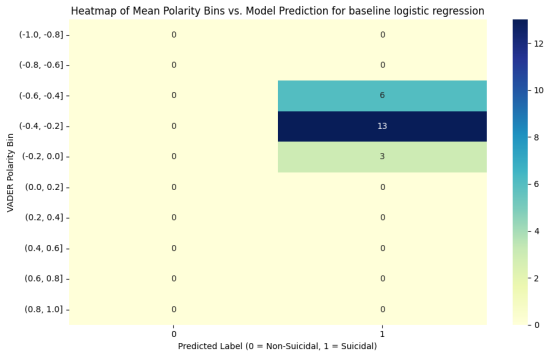



Figure 15: Mean Polarity Scores For Different Predicted Labels from the Baseline Model

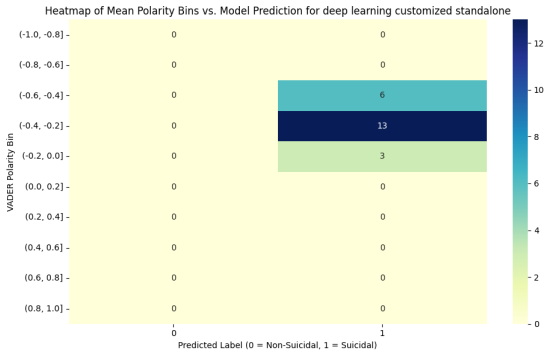



Figure 16: Mean Polarity Scores from the Deep Learning based Model with standalone custom layers

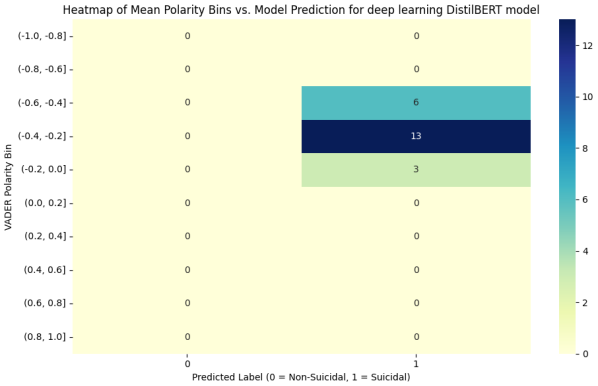



Figure 17: Mean Polarity Scores from the Deep Learning based Model as a whole pipeline

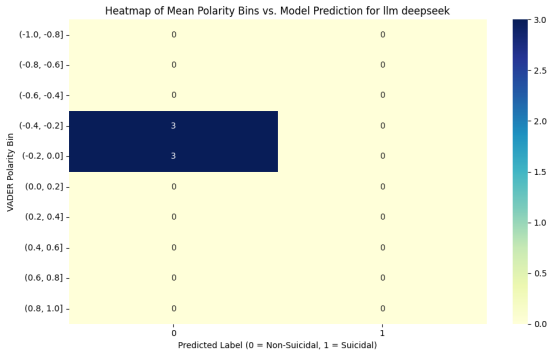



Figure 18: Mean Polarity Scores For Different Predicted Labels from the LLM-based Model

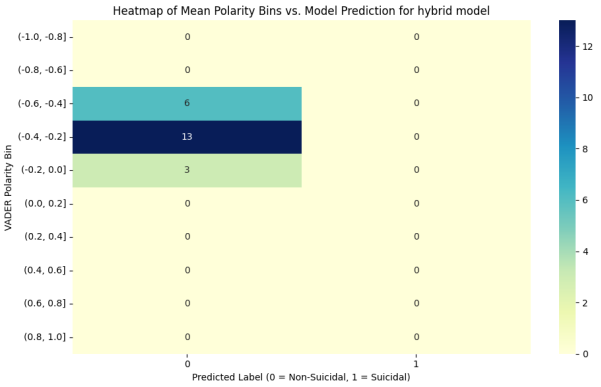



Figure 19: Mean Polarity Scores For Different Predicted Labels from the Hybrid-based Model

**Insights from Polarity Bins** These heatmaps show exactly the ranges of VADER scores that the predictions (from different model respectively) lie in. For example, most records predicted suicidal from the baseline model as well as the deep learning based model (no matter with the entire pipeline or only with custom layers) are with VADER scores between (-0.6, 0.0]. The majority of predictions from the deepseek-based model are non-suicidal, with VADER scores lie between (-0.4, 0.0]. The hybird model mostly predicts texts to be non-suicidal, where their polarities are measured between (-0.6, 0.0]. Therefore, we noticed a majority of posts from the datasets exhibit slightly negative sentiments.

### 5.7.2. Emotion Distributions

Computed by the VADER scores, the average emotion distributions are plotted in the following radar chart. Most of the texts from the Reddit dataset and the Twitter dataset show negative emotions. This is expected because the Reddit dataset captures only the "SuicideWatch" subreddit, and the Twitter dataset contains majorly suicidal posts on Twitter. The depression dataset surprisingly contains most neutral posts. The emotions might tend to negative but not really demonstrating suicidal thoughts. The "Social Media Sentiment Analysis" dataset demonstrate overly positive texts. That is because it is a summary of different kinds of posts (across different platforms) without an intention to gather suicidal ones during data collection. You can see a clearer bar plot as an alternative comparison in the appendix section.

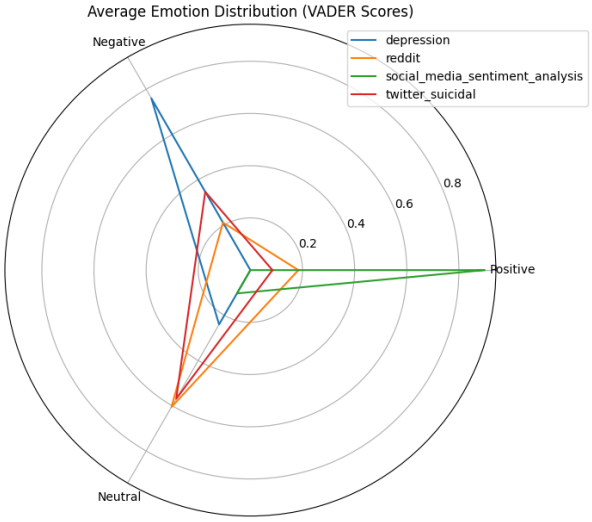



Figure 20: Emotion Distributions of Different Datasets

### 5.7.3. Model Sensitivity

A model's sensitivity is calculated by the sum of the score from texts with a negative sentiment and the rate of predicted suicidal label. The bar plot below shows a comparison of sensitivity scores across different models. It is also a clear visualization showing how the sensitivity is evaluated.

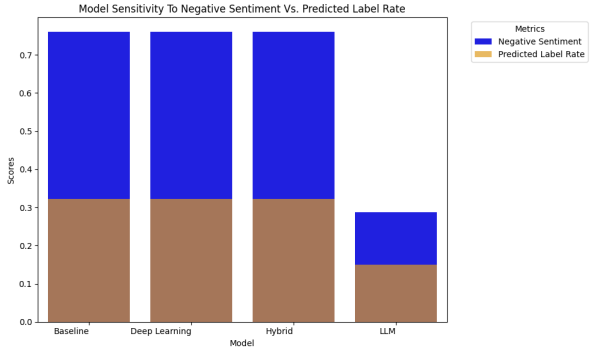



Figure 21: A Comparison Plot of Model Sensitivity

**Insights from Sensitivity Scores** Except the LLM-based model, most models predict posts with a highly positive score, but with a relatively high proportion of negative sentiments. This tells us that the model is more prone to predicting suicidal with posts showing negative sentiments. This is normal because suicidal intentions are normally negative. Surprisingly, the LLM-based model exhibits a lower sensitivity score. This might be the reason of its capability to make suicidal predictions based on the datasets.

### 5.7.4. Subjectivity

The bar plot below compares the subjectivity scores across predictions from different models.

**Insights from the Subjectivity** Except the LLM-based and the hybrid-based model, all other models predict mostly suicidal, from which a majority of texts are subjective. The LLM-based model predicts half of them are subjective. This tells us that the texts are well balanced between subjectivity, among

those that are predicted non-suicidal, because we all know that suicidal texts are highly likely to be subjective and related to one's personal perspective.

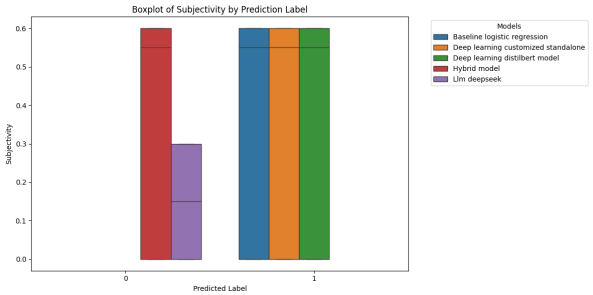



Figure 22: A Comparison Plot of Predictions Subjectivity

## 6. Conclusion

**BERT Models** In response to the research questions, transformers in the BERT family shows a great contribution in suicide detection, mainly because of its capability of preserving contextual meanings from texts. However, we struggled to find an optimal set of hyperparameters (especially the number of epochs) for training. Luckily, we conducted an analysis on also a LLM-based model and a hybrid model. LLMs are proven sophisticated enough to handle textual data. It serves as a good source of comparison. A hybrid model combines the advantages of 2 models of our choice, which includes the baseline model and the DistilBERT-based model. It also balances the tradeoff between overfitting and underfitting, ensuring the model to learn reasonably.

**Linguistics** Lingustic patterns and sentiment markers are clearly analyzed through an *n*-gram analysis, followed by multiple sentiment analysis measures such as the polarity from different scoring metrics, emotion distributions across datasets, a model's sensitivity, and the subjectivity across predictions from different models. It is a sensible setup to understand further how our datasets are balanced and how our model's predictions are biased towards a specific class. In general, we can summarize texts showing suicidal intents have the following properties:

- Negative (or sometimes even polarized) in sentimental interpretation
- Highly subjective

**Bias Mitigation** Reducing biases depends on data collection and pre-processing. For example, the baseline model shows that extracting text embeddings with a pre-trained DistilBERT transformer is more prone to overfitting. This phenomenon might attribute to the fact that the transformer is not yet adapted to our context. Take the fine-tuned DistilBERT as an example. The extracted embeddings yield a more reasonable accuracy. By the way, the LLM-based model used an entirely different tokenizer to extract the text embeddings. That will be another issue which is worth our attention to further investigate how tokenizers affect a model's performance. On the other hand, the source of datasets is also an important factor when it comes to mitigating biases. Like in the Reddit dataset, it has a small number of records, and all of them are within the same class. This is not an ideal dataset for training. Luckily, the training and validation data are resampled prior to the training procedures. Therefore, there are 2 main factors affecting how we mitigate biases: obtaining a good source and using a good tokenizer.

**The Best Model** With our datasets, the baseline model performs the best, probably because it memorizes the patterns from the training data. But in a general suicide detection problem, the hybrid model works the best, whose metrics mostly attain the second highest among other models.

# 7. Future Work

In future extensions of this research, we aim to explore more advanced bias mitigation techniques, such as adversarial training to enhance the ethical fairness across demographic groups. We also propose evaluating the impact of different tokenization strategies on model robustness, particularly how tokenizers affect the detection of nuanced linguistic markers such as sarcasm or slang in suicidal texts. Furthermore, a thoughtful explainability model would be crucial in convincing users, particularly medical experts, of the system's reliability. Two potential directions could be explored: explaining how features within text embeddings contribute to predictions, or investigating how further metadata features (such as the country of a user posting the text) influence a model's predictions. Although the "Social Media Sentiment Analysis" dataset contains some metadata columns, they are not sufficient to split records into reasonable and insightful subgroups. Last but not least, scaling the framework to a real-time detection system and integrating explainability tools like SHAP into deployment pipelines would ensure ethical, trustworthy AI interventions in mental health monitoring.

# 8. Appendix

## Comparison of Models

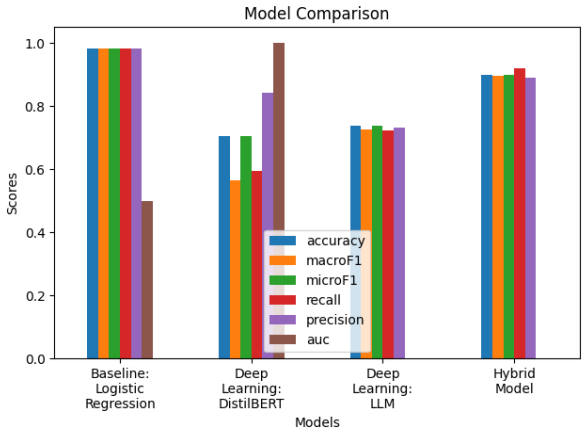



Figure 23: A Summary of Models

## Word Importance

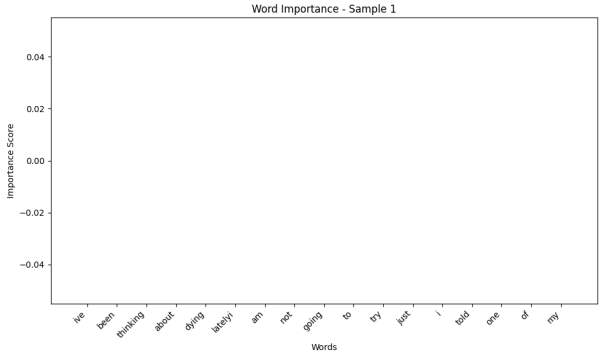



Figure 24: Word Importance in Sample 1

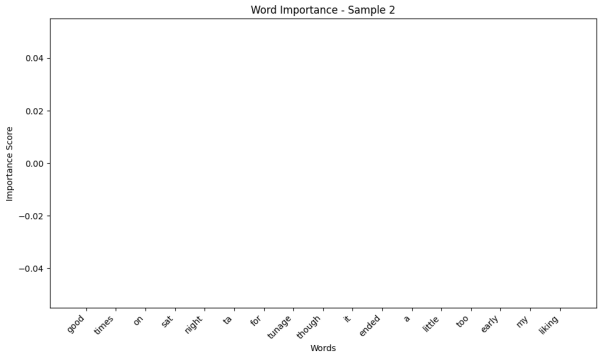



Figure 25: Word Importance in Sample 2

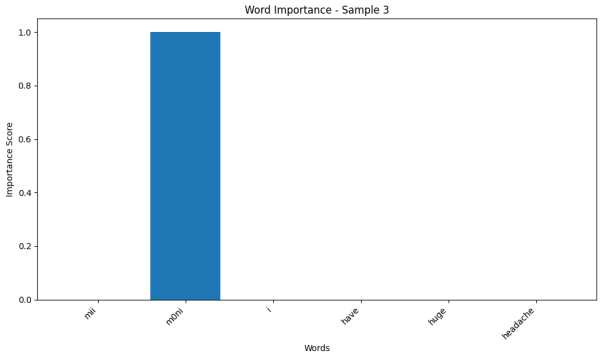



Figure 26: Word Importance in Sample 3

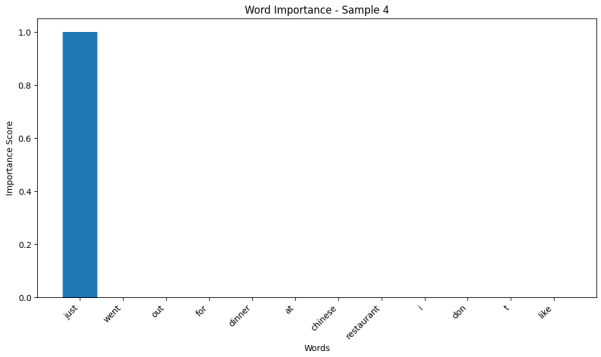



Figure 27: Word Importance in Sample 4

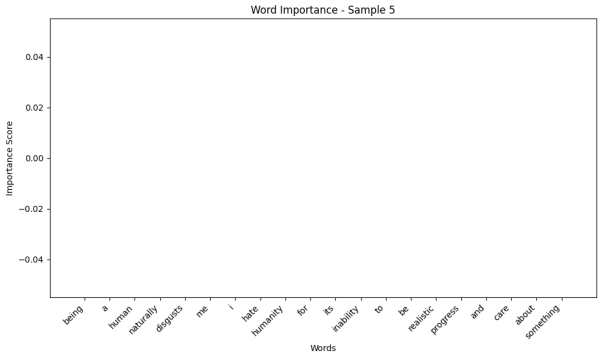



Figure 28: Word Importance in Sample 5

Although some graphs here are empty, it is also important to analyze through them because we can see important words being summarized along the $x$-axis.

## Polarity Distributions

This section shows distributions of the polarity scores calculated by each separate method respectively.

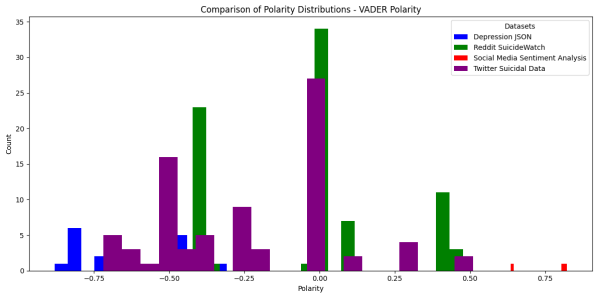



Figure 29: VADER Polarity Score of Word Patterns Across Datasets

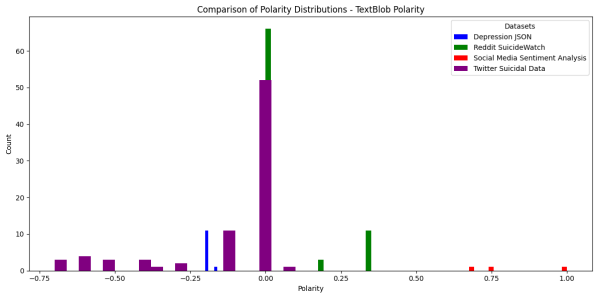



Figure 30: TextBlob Polarity Score of Word Patterns Across Datasets

## Emotion Distributions

This section shows distributions of the average emotion distributions from different datasets.

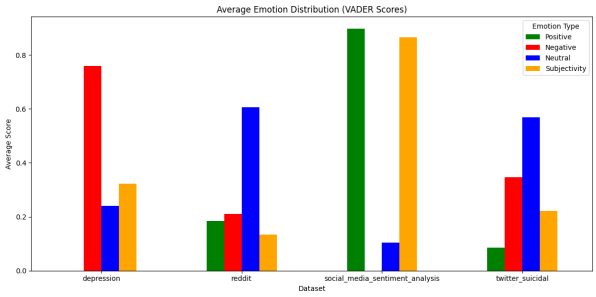



Figure 31: Average Emotion Distributions from Different Datasets

## Details of Models

### Deep Learning based Model

**Table 1**
DistilBERT Base Model

| Layer (type) | Output Shape | Param # |
|---|---|---|
| `distilbert (TFDistilBertMainLayer)` | multiple | 66,362,880 |

**Total params:** 66,362,880 (253.15 MB)
**Trainable params:** 66,362,880 (253.15 MB)
**Non-trainable params:** 0 (0.00 Byte)

**Table 2**
Custom Layers

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_layer_361 (InputLayer) | (None, 2304) | 0 |
| reshape_361 (Reshape) | (None, 3, 768) | 0 |
| bidirectional_361 (Bidirectional) | (None, 128) | 426,496 |
| dropout_361 (Dropout) | (None, 128) | 0 |
| dense_722 (Dense) | (None, 32) | 4,128 |
| dense_723 (Dense) | (None, 1) | 33 |

**Total params:** 430,657 (1.64 MB)
**Trainable params:** 430,657 (1.64 MB)
**Non-trainable params:** 0 (0.00 Byte)

## Acknowledgments

Thanks to the developers of ACM consolidated LaTeX styles https://github.com/borisveytsman/acmart and to the developers of Elsevier updated LaTeX templates https://www.ctan.org/tex-archive/macros/latex/contrib/els-cas-templates.

## References

[1] E. Mohammadi, et al., Detecting suicidal ideation in social media, Journal of Artificial Intelligence Research (2020).

[2] B. H. Sciences, O. A. C. (BHSOAC), Emerging best practices in suicide prevention, 2018. URL: https://bhsoac.ca.gov/sites/default/files/documents/2018-11/Policy%20Brief_Emerging%20best%20practices%20in%20suicide%20prevention_10.17.2018.pdf, accessed: 2025-02-20.

[3] S. R. Braithwaite, C. Giraud-Carrier, J. West, M. D. Barnes, C. L. Hanson, Validating machine learning algorithms for twitter data against suicide rates: A feasibility study, Biomedical Informatics Insights 10 (2018). URL: https://journals.sagepub.com/doi/full/10.1177/1178222618792860. doi:10.1177/1178222618792860.

[4] K. Parmar, Social media sentiments analysis dataset, 2024. URL: https://www.kaggle.com/datasets/kashishparmar02/social-media-sentiments-analysis-dataset, accessed: 2025-02-20.

[5] H. M. Ali, Twitter suicidal data, 2024. URL: https://www.kaggle.com/datasets/hosammhmdali/twitter-suicidal-data, accessed: 2025-02-20.

[6] S. Rajesh, Depression tweets dataset, 2024. URL: https://www.kaggle.com/datasets/senapatirajesh/depression-tweets, accessed: 2025-02-20.

[7] AdSufficient5654, reddit api documentation, n.d. URL: https://www.reddit.com/dev/api/.

[8] Reddit, Reddit data api wiki, n.d. URL: https://support.reddithelp.com/hc/en-us/articles/16160319875092-Reddit-Data-API-Wiki.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2019, pp. 4171–4186.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019). URL: https://arxiv.org/abs/1907.11692.

[11] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019). URL: https://arxiv.org/abs/1910.01108.

[12] Hugging Face, DistilBERT Model Documentation, 2025. URL: https://huggingface.co/docs/transformers/en/model_doc/distilbert, accessed: 21-Feb-2025.

[13] Z. Zeng, X. Ma, Z. Zhou, H. Lin, H. Yu, Y. Zhuang, Z. Fan, Y. Zhang, et al., Ai explainability for all:

A comprehensive survey of post-hoc methods and evaluation, arXiv preprint arXiv:2403.10750 (2024).
[14] E. Mousavi, Nlp series: Day 5 — handling emojis: Strategies and code implementation, 2025. URL: https://medium.com/@ebimsv/nlp-series-day-5-handling-emojis-strategies-and-code-implementation-0f8e77e3a25c.
[15] W. Mousa, How to perform hashtag analysis using natural language processing and machine learning in python, 2023. URL: https://ai.plainenglish.io/how-to-perform-hashtag-analysis-using-natural-language-processing-and-machine-learning-in-python-ea6da817b3a4.
[16] D. Inkpen, n.d. URL: https://www.site.uottawa.ca/~diana/csi5180/StopWords.
[17] DeepSeek-AI, DeepSeek-VL: Multimodal Large Language Model, 2024. URL: https://github.com/deepseek-ai/DeepSeek-VL/tree/main, accessed: 2025-02-20.
[18] Hugging Face, Autotrain text classification parameters, https://huggingface.co/docs/autotrain/text_classification_params, 2024. Accessed: 2025-04-21.
[19] Y. Nagaraj, F1 score, Math and Core Machine Learning (2023). URL: https://www.linkedin.com/pulse/f1-score-yeshwanth-n/.
[20] Pete Keen, Matthew Honnibal, Roman Yankovsky, David Karesh, Evan Dempsey, Wesley Childs, Jeff Schnurr, Adel Qalieh, Lage Ragnarsson, Jonathon Coe, Adrián López Calvo, Nitish Kulshrestha, Jhon Eslava, @jcalbert, Tyler James Harden, @pavelmalai, Jeff Kolb, Daniel Ong, Jamie Moschella, Roman Korolev, Ram Rachum, Romain Casati, Evgeny Kemerov, Karthikeyan Singaravelan, John Franey, Textblob, n.d. URL: https://textblob.readthedocs.io/en/dev/.
[21] Tom Aarsen, Joel Nothman, Steven Bird, Alexis Dimitradis, Danny Sepler, Dmitrijs Milajevs, Francis Bond, Ilia Kurenkov, purificant, Liling Tan, Dan Garrette, Peter Ljunglöf, Mikhail Korobov, Alex Rudnick, Edward Loper, Ewan Klein, Trevor Cohn, Pierpaolo Pantone, Nathan Schneider, Álvaro Justen, Morten M. Neergaard, Nltk - vader documentation, n.d. URL: https://www.nltk.org/api/nltk.html.

## A. Online Resources

The source codes for this project and ceur-art style are available via the following links respectively:

- GitHub,
- Overleaf template.