CSC 396 Final Project:
Detecting Mental Health Distress From Reddit Posts

Team Members

Haocheng Cao
Kyle Dervin
Rena Wang

Instructor

Mihai Surdeanu

Date

Dec 9, 2025

Introduction

This project investigates whether machine learning models can classify mental-health–related Reddit posts into three categories: SuicideWatch, depression, and teenagers. The task is to detect emotional distress and potential suicidal intent indicated in online posts, using both a classical approach (to serve as a baseline for comparison) and a transformer-based architecture.

Motivation

Social media platforms often serve as informal emotional support spaces, especially for young people who may not feel comfortable getting access to traditional mental health detection and treatments. Many users may unknowingly disclose underlying mental health struggles through their online posts. This presents an opportunity for our machine learning project to proactively identify and flag high-risk content, enabling timely intervention and outreach efforts.

Project Goals

We aim to build and compare:

- A baseline model using TF-IDF features and logistic regression.
- A transformer-based classifier using DistilBERT.

We will then evaluate these models, identify their strengths and weaknesses, and perform qualitative error analysis to understand where the models are error-prone and potential sources of future improvement.

Connection to Course Goals

This project reflects the course themes of:

- Curiosity & research mindset: exploring a meaningful, real-world dataset.
- Community & engagement: iterative project design, collaborative decision making.
- Responsible AI: awareness of model limitations in sensitive domains.

Data Source and Description

We used the Kaggle dataset “Suicide and Depression Detection”, containing Reddit posts from three subreddits: SuicideWatch, depression, and teenagers (Source). This dataset is a csv file with 2 string columns. The first column is the actual text from each individual reddit post (one per row). The second column is a class for the associated text with three possible labels “SuicideWatch”, “depression”, and “teenagers”. The distribution of labels was perfectly even, with just over 116,000 posts per label. After filtering, the dataset provides several thousand posts, which we split into train and development (dev) sets using an 80/20 split with a fixed random seed. The label “SuicideWatch” consists of posts that express suicidal ideation, crisis, or urgent distress. The label “depression” consists of posts that describe depressive symptoms, hopelessness, or emotional pain. The label “teenagers” consists of posts referencing adolescence, school life, and general teen issues. There are a few challenges with these labels. For one, the emotional language overlaps heavily between categories, especially between the “SuicideWatch” and “depression” categories. Furthermore, many teenager posts may overdramatize and use strong language more commonly seen in a more-crisis prone label category. Posts can describe all sorts of mixed topics making it difficult to pick up on distressing signals. Taking data from subreddits results in a significant amount of noise, and may not be truly reflective of the mental state of the people posting.

Dataset Preprocessing

Preprocessing consisted of first normalizing the text. All text was made lowercase, end of line characters and escape characters were removed, whitespace was stripped, and spaces and punctuation were made consistent throughout. Each label was mapped to an integer. There were just a few rows with NaN values in the class column so these were dropped. Train/test split was used to have 80% of the data used for training, and the remaining 20% stored away until test time.

Core Implementation

The first model that was introduced was a baseline model for future comparison. We used a TF-IDF model in combination with logistic regression. TF-IDF stands for Term Frequency-Inverse Document Frequency. It converts text into numerical feature vectors for a machine learning model to process. Every word seen becomes a feature. Its purpose is to highlight words that are important in a document (give those words more weight) but not too common in the larger language. The idea here was that words like “depressed” or “suicidal” might be common in posts from higher-risk individuals, but not common across all posts. This model would thus give more weight to these weights and less to more common filler words that may still appear frequently, but also appear frequently in all posts, regardless of the label. In order to gain back some of the context lost by unigrams (single words only), we also included bigrams (which can account for negation or common word combinations). The vocabulary limit was set to only the most frequent 20,000 unigrams and bigrams in order to prevent overfitting. Words that appeared in fewer than 5 documents were removed in order to reduce some noise. Logistic regression (and in particular, multinomial logistic regression) is just a simple linear classification model, and it works well with sparse, high-dimensional data like that produced from the TF-IDF features. It also performs surprisingly well on short-text tasks. Since high-dimensional datasets can take longer to converge, the maximum iterations for training were set to 1000.

Our second model was implemented in hopes of seeing improved performance in comparison with the baseline. The hope was that a transformer network would be able to better capture and model both context and long-range dependencies in the sequential text data, which is lacking in the baseline model. To represent the inputs, we did Tokenization via DistilBertTokenizerFast with truncation and padding. The max token length of 128 was set in order to save memory and speed up training. We used DistilBertForSequenceClassification, which is a pre-trained DistilBERT model adapted for sequence classification tasks. We also used a schedule for the learning rate that linearly increases it at first and then decreases it over training. In addition, we used a version of the Adam optimizer with weight decay that is commonly used in transformers. We changed the base model to have a classification head with 3 logit outputs instead of 2. We also set the number of epochs through the training data to 3.

Results and Comparative Evaluation

Baseline (TF-IDF model) Results:

- Accuracy: 0.82
- Macro-Precision: 0.81
- Macro-Recall: 0.82
- Macro-F1: 0.81

Baseline Results Summary:

The logistic regression classifier performs consistently across classes, with relatively balanced precision and recall. For a simple baseline using TF-IDF features, this is strong performance, given the noisy nature of subreddit-derived labels.

Performance by Label:

Teenagers

Precision	Recall	F1
0.89	0.93	0.91

This class performs the best by a significant margin. Having a very high recall shows the model almost always identifies a teenager-labeled post correctly. This can be explained by the distinctive lexical cues in this class, such as references to school, parents, age, friends, or homework. These references provide more emotional content rather than complicated clinical mental health language.

SuicideWatch

Precision	Recall	F1
0.78	0.76	0.77

This class is slightly limited compared to Teenagers. Crisis posts often contain subtle expressions of suicidal ideation (e.g., “I don’t want to be here anymore”), which may not include explicit keywords. Without contextual modeling, TF-IDF often confuses subtle crisis language with severe depressive language. The precision and recall in this suggest balanced errors: False positives are depressive posts with strong hopelessness, and false negatives are implicit suicidal ideation lacking explicit terms.

Depression

Precision	Recall	F1
0.77	0.75	0.76

This class has the lowest performance. Depression posts share vocabulary with both teenagers and SuicideWatch, leading to 2 cases. False positives are distress expressed by teenagers classified as depression, and false negatives are crisis-level posts misclassified as general depression. In here, the emotional language is broad and varied (e.g., “sad”, “empty”, “can’t get out of bed”), and without deeper contextual understanding, this model struggles to distinguish depressive symptoms from general distress or crisis language.

DistilBERT Results:

Epoch	Train Loss	Dev Accuracy	Macro Precision	Macro Recall	Macro F1
1	0.3888	0.8515	0.8545	0.8515	0.8519
2	0.3124	0.8601	0.8598	0.8601	0.8598
3	0.2790	0.8603	0.8607	0.8603	0.8604

DistilBERT Results Summary:

Compared to Baseline, DistilBERT improves macro-F1 by about 5 percent. We could see a stronger gain on SuicideWatch due to improved contextual understanding of the model. The smooth drop in train loss suggests a good convergence and avoidance of significant overfitting. The high macro F1 score shows balanced performance across three classes, giving better outputs in general.

Results Comparison & Observations

Model	Accuracy	Macro Precision	Macro Recall	Macro F1
Baseline	0.82	0.81	0.82	0.81
DistilBERT	0.8603	0.8607	0.8603	0.8604

Accuracy improved by +0.04 (82% → 86%). Macro-F1 improved by +0.05 (0.81 → 0.86). Macro-Recall improved by +0.04. Macro-Precision improved by +0.05. The transformer model saw improved distinguishing between SuicideWatch vs depression. It also displayed an improved understanding of long, narrative posts and better handling of ambiguous “teenagers” posts. The TF-IDF logistic regression baseline performs well on posts with clear lexical cues, especially for the “teenagers” class. However, it struggles to distinguish between “depression” and “SuicideWatch”, where contextual understanding is essential. DistilBERT addresses this deficiency by modeling semantic relationships across entire posts, allowing it to detect subtle indications of crisis severity, emotional escalation, and implicit suicidal ideation.

In the notebook, we created a radar chart to visualize how DistilBERT dominates on all four metrics. We also plotted side-by-side confusion matrices, which show how the confusion matrix does a good job on Teenagers but frequently misclassifies SuicideWatch and depression. The DistilBERT matrix reduces these cross-class errors, especially for ambiguous posts and longer narratives.

Error Analysis

To better understand the model behavior, we pick the DistilBERT model to do error analysis, since it's the better-performing one.
We collected predictions on the dev set and built a DataFrame with text (decoded input text), gold (gold label), and pred (predicted label). We then filter to mismatches of gold and pred. There are 9727 such misclassified examples.

Common Patterns

- SuicideWatch labeled as depression
    - Many crisis posts express depression without explicit suicidal keywords.
- Teenagers labeled as depression
    - Teenagers like to use sarcastic language or dramatic jokes to show their feelings; many of them can be classified as depression, although the meaning could actually be the exact opposite.
- Label noise
    - There are posts heavy in suicidal phrasing but labeled as depression in the dataset. Here, the model arguably makes the more clinically reasonable choice (SuicideWatch), revealing label noise rather than a true model error.

Representative Examples

1) "I'm ready!Haha yay! …
   Please be happy. I hope joy and success for you. I wish you get through the hard days. I couldn't."

- SuicideWatch but predicted depression
- This post uses positive or neutral language at the beginning, which is very misleading. The only part that hints the suicidal ideation is the last part: "I couldn't", but it doesn't show a strong tone, which is what the transformer relies heavily on.

2) "I am cripplingly addicted to custard Edit: r/custard wouldn't take me so I come here for salvation.
   … I started buying it almost daily. The kids would some nights go hungry because daddy needs his custard.

… It's started effecting my marriage. My wife would sometimes find me asleep naked covered in Paul's goodness.
… This is a cry for help.
Please
Save me."

- Teenagers but predicted depression
- For a human being, this is for sure a joke written in sarcastical language, but for the model, the only thing they would capture is the high-risk emotional deterioration because of the "crying for help" at the end. Also, the writing style doesn't resemble typical teenager, as there are keywords like "marriage" and "kids". This is an example of an inconsistent and noisy example.

3) "Just 2 hours left :) My parents are going to work in 2 hours and I will put an end to the misery and pain that I have held within me. I have made this account in hopes of finding out a way to save myself and my family from my depression, but it hasn't worked out. I already have the rope tied to my fan. Bye :)"

- Depression but predicted SuicideWatch
- This example appears to be a noise in the dataset rather than a classifier mistake. The post clearly expresses both depressive symptoms and suicidal intent, with the latter being much stronger. It is reasonable for the model to classify the text as SuicideWatch. This kind of label noise can negatively impact model performance and make evaluation less reliable.

Text Length Analysis

We explicitly analyze the effect of text length on missed SuicideWatch cases by creating an analysis DataFrame with fields: class, pred_transformer, text_len, label_name, and pred_name. We define suicide_missed as those that are SuicideWatch but predicted not as the same class. Here are the data:

- Total SuicideWatch cases: 23,208
- Correctly identified: 19,071
- Missed (False Negatives): 4,137

We then plot KDE distributions of text length for correctly detected vs. missed SuicideWatch posts. Here are the observations:

For long texts (especially above ~2,000 characters), the density of missed SuicideWatch cases is noticeably higher than that of correctly detected ones. We manually inspect the three longest missed examples (lengths ~17k–22k characters) and find a consistent pattern:

- The first part is a long story about life history and depression.
- The explicit suicidal intent or plan (e.g., “after I kill myself…”, “this is my final message”) appears near the end of the post.
- Since DistilBERT truncates at 128 tokens, the model never “sees” the conclusion. It effectively reads only the introduction and therefore often predicts depression instead of SuicideWatch.

This is the lost ending truncation problem and is a major source of SuicideWatch false negatives in our model.

Semantic Overlap Analysis

Both SuicideWatch and depression classes share a highly overlapping vocabulary, including words like “sad”, “tired”, “worthless”, “pain”, “empty”, or “hopeless”. Some SuicideWatch posts are essentially “depression posts + one sentence of suicidal intent”. The confusion matrices show a substantial crossover, where many depression posts are predicted as SuicideWatch, and vice versa. This shows that the semantic ambiguity and label noise are the challenges we need to overcome if we want to improve the project in the future.

Sentiment Hypothesis

By using TextBlob, we approximate sentiment polarity for SuicideWatch posts. Here are the results:

- Correctly detected SuicideWatch: average sentiment ≈ –0.0082 (slightly negative).
- Missed SuicideWatch (False Negatives): average sentiment ≈ +0.0052 (slightly more positive or neutral).

These results weakly support the hypothesis that more “neutral/positive” framing can cause the model to miss crisis intent.

Limitations & Future Work

Data Limitations:

This project has several limitations in the data itself. Because labels are assigned based on the subreddit where each post was originally submitted, the data contains a lot of label noise. There are examples of posts in the depression subreddit showing explicit suicidal intent, and some posts in SuicideWatch contain general emotional distress. This overlap reduces the reliability of the ground truth and can cause models to be “incorrect” even when they make reasonable predictions.

Model Limitations:

There are a few limitations with our modeling approach. First, we set an input length limit of 128 tokens, which causes the longer posts to be truncated, and in many SuicideWatch examples, the most critical information about suicidal intent always appears near the end of the post. This causes the model to always read in the depressive backstory and never see the explicit plan or final decision, which leads to false negatives where true SuicideWatch posts are predicted as depression. Second is the ambiguity between the depression and SuicideWatch classes. They both have similar vocabulary of suffering, which causes the model to rely on subtle cues or a few trigger phrases to distinguish them. Lastly, our modeling scope is also limited to the baseline and transformer architecture, with relatively minimal hyperparameter tuning due to computational constraints. We did not conduct extensive sweeps over learning rates, batch sizes, or regularization, so it's likely that better configurations and alternative architectures could have a stronger performance.

Future Directions:

There are several promising directions we would like to explore. The first thing is to mitigate the truncation problem by adopting models or strategies that can handle longer contexts. We could try Longformer or BigBird, which could process full-length posts without cutting off crucial endings, and use a simpler sliding-window strategy, passing overlapping segments of a long text through DistilBERT and then aggregating the segment-level predictions. Then, we could tackle the label noise by incorporating weak supervision or semi-supervised learning, or by constructing a smaller, manually curated validation set with more reliable "gold" labels. Another direction is to reframe the task from three-way classification to a binary risk detection problem, where the main distinction is between high-risk crisis posts (SuicideWatch) and non-crisis posts (depression and teenagers combined). This would align the model more closely with real-world triage goals, where recall on truly high-risk cases is more important than fine-grained distinctions between non-crisis categories. Also, we could explore additional features beyond raw text, such as posting frequency, temporal patterns, or discourse structure, while carefully considering privacy and ethics.

Reflection

Throughout the project, we gained hands-on experience with both classical and transformer-based NLP models and learned why transformers often outperform linear baselines on complex text. Dealing with messy, noisy real-world data made us observe that dataset issues like label noise and truncation can matter as much as model choice. We waited a lot of time for the dataset to process and train, which we should probably get used to in real working environments. Most importantly, working on a sensitive domain like mental health forced us to think beyond raw metrics. A model with 86% macro-F1 can still miss thousands of high-risk posts. Through detailed error analysis, we started to "read" model failures instead of just collecting scores. Understanding why a model fails is one of the most valuable lessons we learned from the class and the project.