CS5624 - Natural Language Processing

Project Report - Group 15

Varun Ganta (111901051), Deon Saji (111901022) & Naren Loganathan (111901056)

Suicide Prevention - Detection of Suicidal Posts and Messages

Introduction

Around 800,000 people die from suicide every year. By the time you finish reading this report, an additional dozen people around the world will have likely taken their lives. Unfortunately, suicide is also one of the leading causes of death in the younger age demographic.

When a person has intentions of committing suicide or self harm, there are chances of such thoughts manifesting in their messages and posts. Being able to detect / notice signs of suicidal intent early on (along with timely intervention) can help prevent suicides. We aim to estimate the chances of an individual having suicidal intent by analysing their activity (i.e. messages and comments) on social media platforms.

Our problem statement can be rephrased as follows in the context of NLP:

> ‘Design and test models that classify and separate messages hinting at suicide / depression from others’

Difficulties

By looking at words commonly used in messages with suicidal intent, it may seem like such messages stand out quite a bit from others.

For instance, a message containing the word ‘suicide’ is likely to be a positive (i.e. a message hinting at suicide / depression). However, sentences like ‘XYZ is watching the Suicide Squad movie.’ definitely don’t fit within the same category.

Messages displaying suicidal intent are often fairly subtle. For instance, take a look at the below sentences:

- I don’t want to live anymore.
- I want to end it all.

It’s probably easy for us humans to identify that something is not right here, but it might be difficult for a model built on very naive assumptions to figure out that these are in fact positives, and not negatives.

Another thing that is somewhat unsurprising is that it is generally quite difficult to distinguish between messages with genuine suicidal intent and messages that reflect feelings of anxiety & depression (the models we tried for filtering between these didn't perform very well).

Dataset

We decided to delve into Reddit (a large social network comprising various communities) for our dataset. Ideally, we needed a bunch of messages conveying suicidal intent (positives) as well as many messages that were about more lighthearted things (negatives).

The subreddit 'r/SuicideWatch' was a place for people struggling with suicidal thoughts, an good place to collect data. *Addressing privacy concerns:* In general, a rule within the community is to make posts under totally anonymous accounts. Additionally, multiple other studies in the same area have used Reddit for their datasets, so we went ahead with our choice of dataset. All of the information we scraped is publicly available.

Likewise, we scoured the 'r/Anxiety' and 'r/depression' subreddits for similar samples.

For negative (i.e. messages having nothing to do with suicide) examples, we scraped data from the 'r/movies', 'r/books', 'r/popular' and 'r/Jokes' subreddits.

For the purpose of training, we concatenated the title and contents of the Reddit posts and fed it into our models.

What we did

The successful outcome:

After scraping and getting our dataset, we cleaned / preprocessed it (the first step in almost every NLP task). The major preprocessing steps we tried were removal of punctuation and special characters, converting words to lowercase, word lemmatization and removal of stop words, courtesy of the NLTK library.

We implemented two models - one using an LSTM network, and the other using a Naive Bayes Classifier. We decided to compare them on the basis of performance.

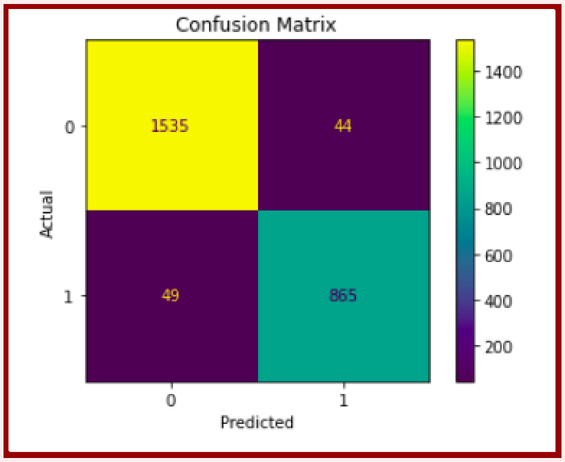
The input data needs to be converted into a vector and therefore a TFIDF vectorizer is used in the case of the Naive Bayes Classifier. In the case of the LSTM model, a tokenizer from tensorflow.keras is used.

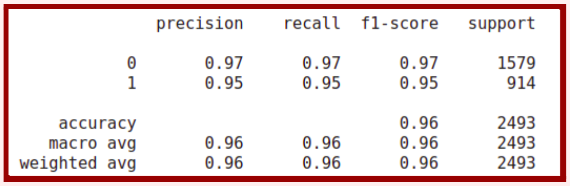
We had a training-test split of 70%-30% (with random shuffling) for both models with a total dataset size that was in the ballpark of 10,000 examples.

The way we labelled / annotated our dataset was as follows:

Positive (1)	Negative (0)
Posts from the following subreddits: - r/SuicideWatch - r/Anxiety - r/depression	Posts from the following subreddits: - r/movies - r/books - r/popular - r/Jokes

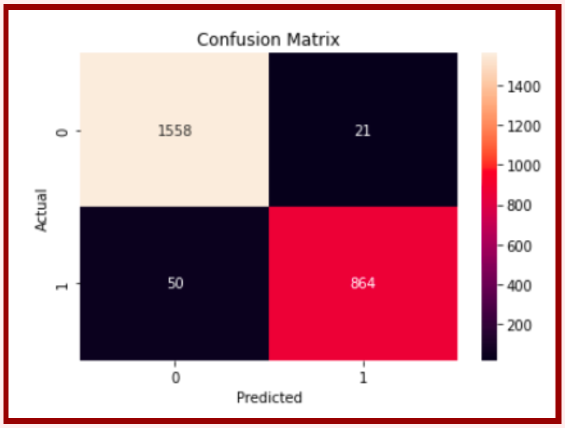
Results on the test set for the Naive Bayes Classifier:

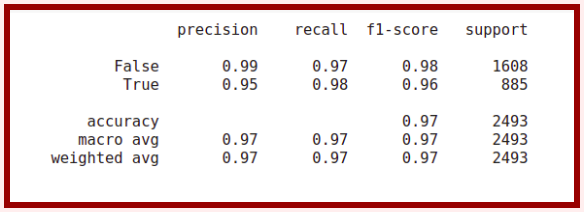




```
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      1579
           1       0.95      0.95      0.95       914

    accuracy                           0.96      2493
   macro avg       0.96      0.96      0.96      2493
weighted avg       0.96      0.96      0.96      2493
```

Results on the test set for the LSTM network:



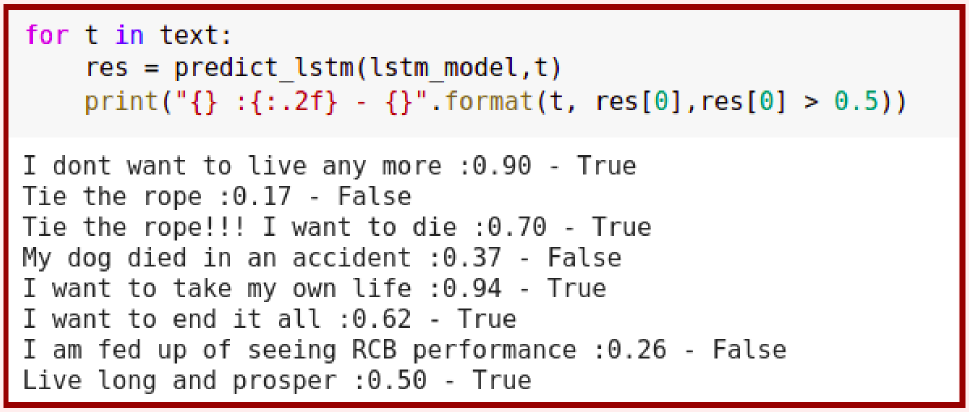


```
              precision    recall  f1-score   support

       False       0.99      0.97      0.98      1608
        True       0.95      0.98      0.96       885

    accuracy                           0.97      2493
   macro avg       0.97      0.97      0.97      2493
weighted avg       0.97      0.97      0.97      2493
```

As you can see, we were able to achieve very high F_1 scores on both models. The score achieved in the case of the LSTM model was slightly better, although this isn't sufficient to claim that it is superior to the NB Classifier (as different shuffles were used for the training-test splits).

Below are some of the sentences we tested on both models. In the case of the LSTM model, we also print out the confidence score on the side (if it exceeds 0.5, we set the flag as 'True', and 'False' otherwise).

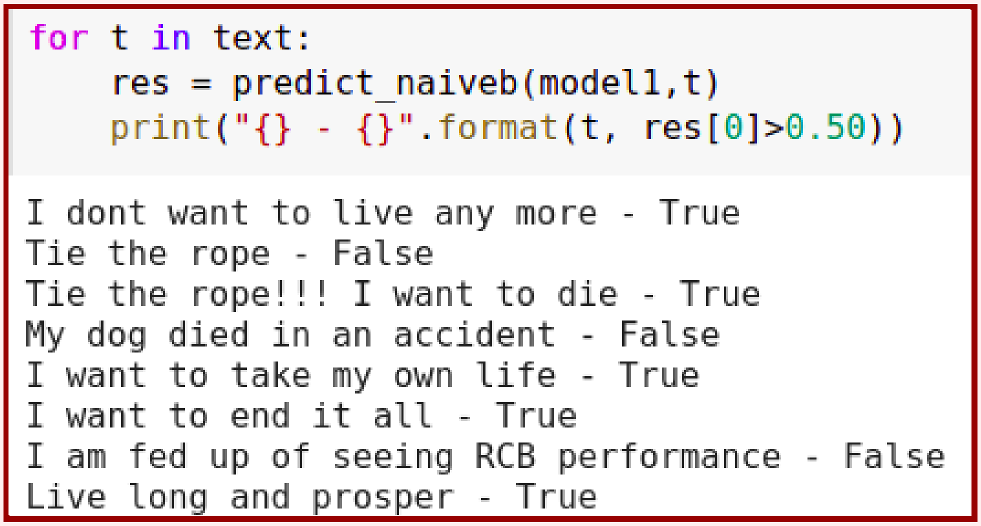
LSTM Model



```
for t in text:
    res = predict_lstm(lstm_model,t)
    print("{} :{:.2f} - {}".format(t, res[0],res[0] > 0.5))
```

```
I dont want to live any more :0.90 - True
Tie the rope :0.17 - False
Tie the rope!!! I want to die :0.70 - True
My dog died in an accident :0.37 - False
I want to take my own life :0.94 - True
I want to end it all :0.62 - True
I am fed up of seeing RCB performance :0.26 - False
Live long and prosper :0.50 - True
```

Naive Bayes Classifier



```
for t in text:
    res = predict_naiveb(model1,t)
    print("{} - {}".format(t, res[0]>0.50))
```

```
I dont want to live any more - True
Tie the rope - False
Tie the rope!!! I want to die - True
My dog died in an accident - False
I want to take my own life - True
I want to end it all - True
I am fed up of seeing RCB performance - False
Live long and prosper - True
```

Here we observe that some of the problematic sentences we identified in the 'Difficulties' section earlier are being identified correctly by both models.

Interestingly, 'Tie the rope' is flagged as false by itself. However, when we add some extra context, e.g. 'I want to die.' then it gets assigned a positive label. Sentences involving the death of other individuals / subjects are also generally identified correctly with a negative label.

Amusingly, the sentence ‘Live long and prosper.’ seems to have been misclassified by both models (as a false positive), although it is worth noting that the confidence score of the LSTM model is almost 0.5 flat (so the positive label was assigned with barely any confidence). This could be due to the fact that the word ‘live’ is mentioned quite often in posts pertaining to suicide, anxiety and depression, whereas it isn’t seen all that often in messages / posts pertaining to more light-hearted topics. If anything, our training data could have been a bit more comprehensive.

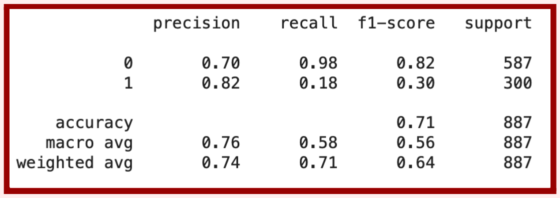
That being said, if there are misclassifications, it is better to have more false positives as opposed to false negatives. This is because false positive ⇒ you accidentally identify a normal individual as having suicidal intent, whereas false negative ⇒ a person having suicidal intent goes undetected by the model (which is way worse and goes against our goals - detection is crucial!).

An unsuccessful outcome:

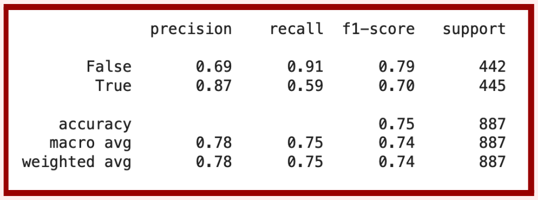
Prior to the successful outcome, we attempted to distinguish between messages / posts hinting at suicide (1) versus anxiety and depression (0).

Positive (1)	Negative (0)
Posts from the following subreddits: - r/SuicideWatch	Posts from the following subreddits: - r/Anxiety - r/depression

We attempted to train an LSTM model with the same architecture and a random forest classifier using these datasets alone. However, the test set validation was a big miss, and we got pretty poor F_1 scores.



```
              precision    recall  f1-score   support

           0       0.70      0.98      0.82       587
           1       0.82      0.18      0.30       300

    accuracy                           0.71       887
   macro avg       0.76      0.58      0.56       887
weighted avg       0.74      0.71      0.64       887
```



```
              precision    recall  f1-score   support

       False       0.69      0.91      0.79       442
        True       0.87      0.59      0.70       445

    accuracy                           0.75       887
   macro avg       0.78      0.75      0.74       887
weighted avg       0.78      0.75      0.74       887
```

Left: Random Forest Classifier, Right: LSTM Model

In particular, note the very poor recall & F_1 scores on the positive label (i.e. a majority of the posts indicating suicidal intent are being misclassified, which isn’t good at all).

This only further highlights the challenge of separating suicidal messages from those that have undertones of anxiety and depression.

References

Naive Bayes Classifier in Tensorflow

Naive Bayes Scikit Learn

Code Reference 1

Code Reference 2

Towards Datascience Suicidal Classifier

Towards Datascience TF-IDF

Towards Datascience Goodbye World

Supervised Learning for Suicidal Ideation Detection in Online User Content

Learning Models for Suicide Prediction from Social Media