Depression & Suicide Detection in Reddit Posts

Vasilescu Andreea
Diaconescu Vlad-Eduard

University of Bucharest
Faculty of Mathematics and Informatics
Masters of Natural Language Processing

June 2023

Table of Contents

1. Tools and Methods Used
2. Theme, Importance and Motivation
   - Motivation
3. Experiment
   - Data-Set
   - Pre-processing
   - Named Entity Recognition
   - Bag of Words
   - SVM
   - NN
   - BERT
4. Comparison Between Methods used
5. Concluzii finale

Table of Contents

6 Future Work

Tools and Methods Used

- **Google Colab**
  - accessible from any device - we upgrade to Google Collab Pro, as we needed more GPU - good for collaborating - easy to work with
- **Python**
  - perfect for NLP
  - offer lots of frameworks and libraries
  - offer flexibility
  - lots of online information

Theme

- Our project theme is Suicide and Depression Detection in Reddit posts.
- The data-set was available on Kaggle
- We aimed to classify messages into one of the following categories: suicide or non suicide

Bio-Medical NLP NER AI binary classification NN SVM BERT

Importance

- Depression is impacting more and more people every day, resulting in a decreased quality of daily life.
- For many who face this problem, the internet is the place to vent and express their feelings
- Every year, arround 700.000 people are commiting suicide.

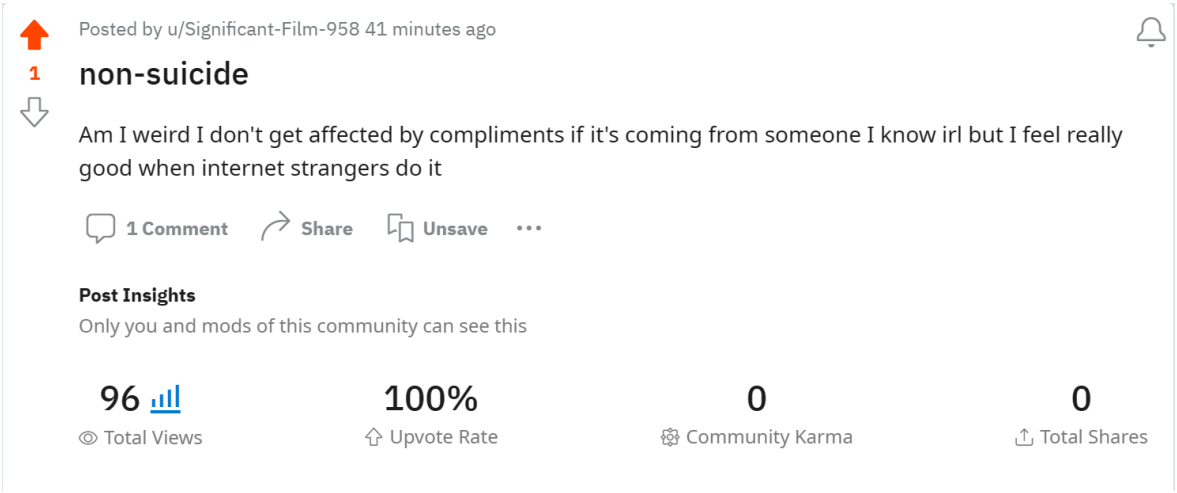
Montivation

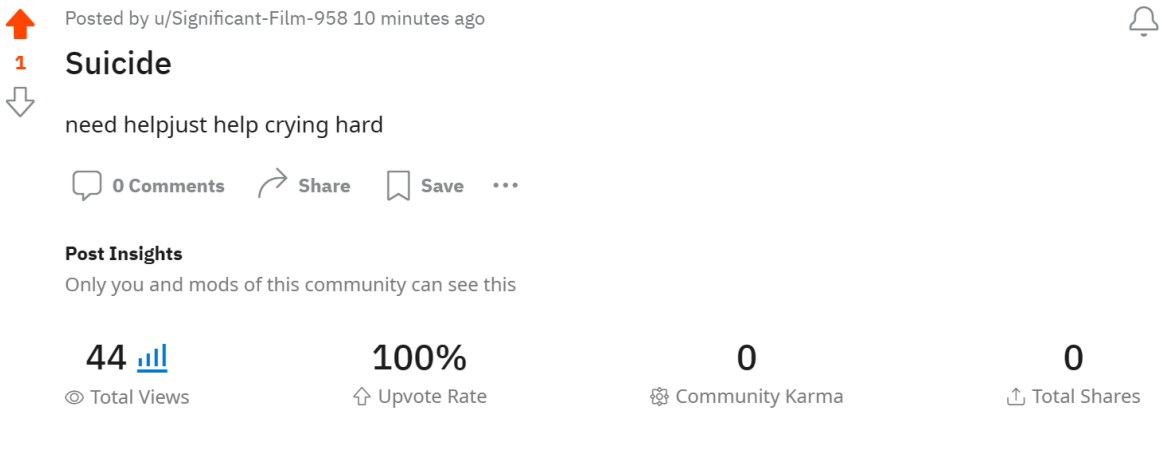
- Relevance of the theme
- Even social media itself is a cause of depression, so the least we can do is to automatically identify those who are going trough tough times
- Need to help those in need, by identifying their depressive thoughts on social media

About data set

- Data set was collected from Kaggle
- For each Reddit proposed for the classification, the text and its class was provided. (suicide or non suicide)
- CSV format
- The number of data provided was 232075, but we only used 33153

Data Set


Posted by u/Significant-Film-958 41 minutes ago
1
non-suicide
Am I weird I don't get affected by compliments if it's coming from someone I know irl but I feel really good when internet strangers do it
1 Comment
Share
Unsave
Post Insights
Only you and mods of this community can see this
96
Total Views
100%
Upvote Rate
0
Community Karma
0
Total Shares


Posted by u/Significant-Film-958 10 minutes ago
1
Suicide
need helpjust help crying hard
0 Comments
Share
Save
Post Insights
Only you and mods of this community can see this
44
Total Views
100%
Upvote Rate
0
Community Karma
0
Total Shares

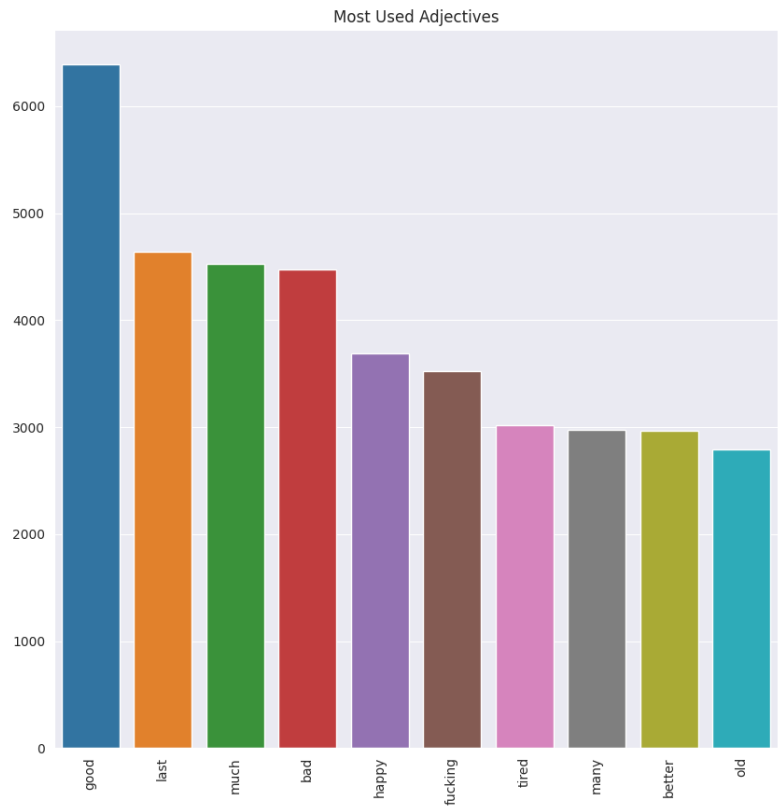
Pre-processing

We split the data info Split the data into reddit_paths, reddit_data and reddit_classes using pandas library.
We proceed with the following:

- stop words, using NLTK for english vocabulary
- lower case
- remove white spaces
- remove numbers
- remove special characters
- remove email addresses
- lemmatization
- stematization
- tokenization

NER

We have splitted our data into suicide and non-suicide, as we wanted to see which are the most used adjectives based on category and overall.
We have used Spacy, the popular open source library for NLP.



NER on category

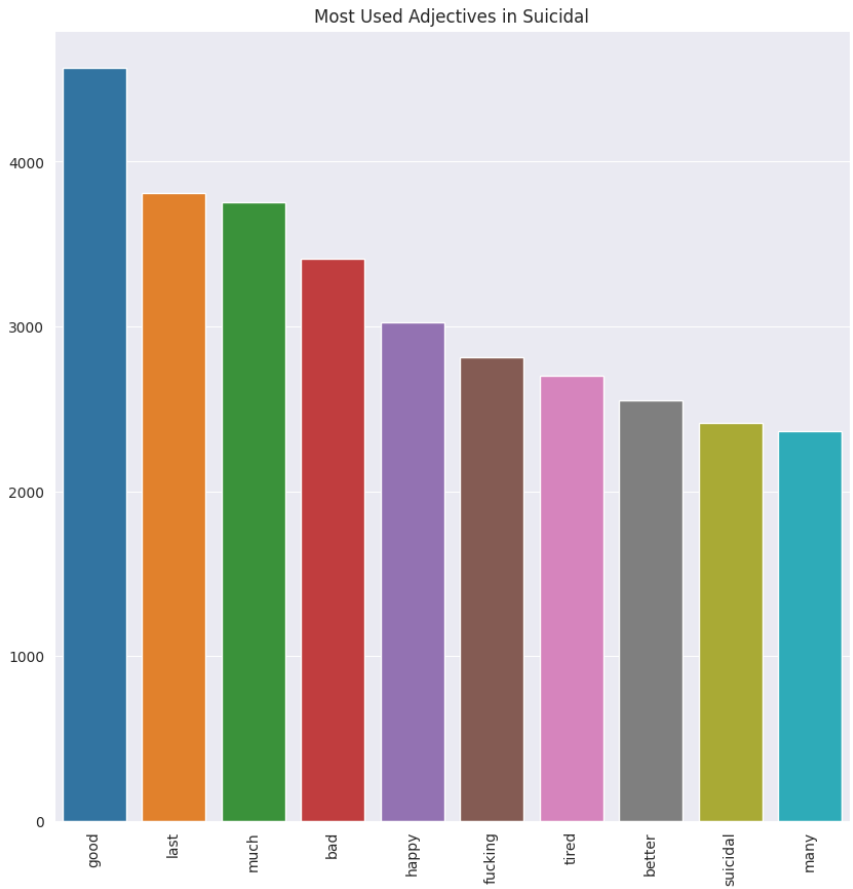


Figure: Most Used Adjectives on Reddits classified as suicidal

NER on category

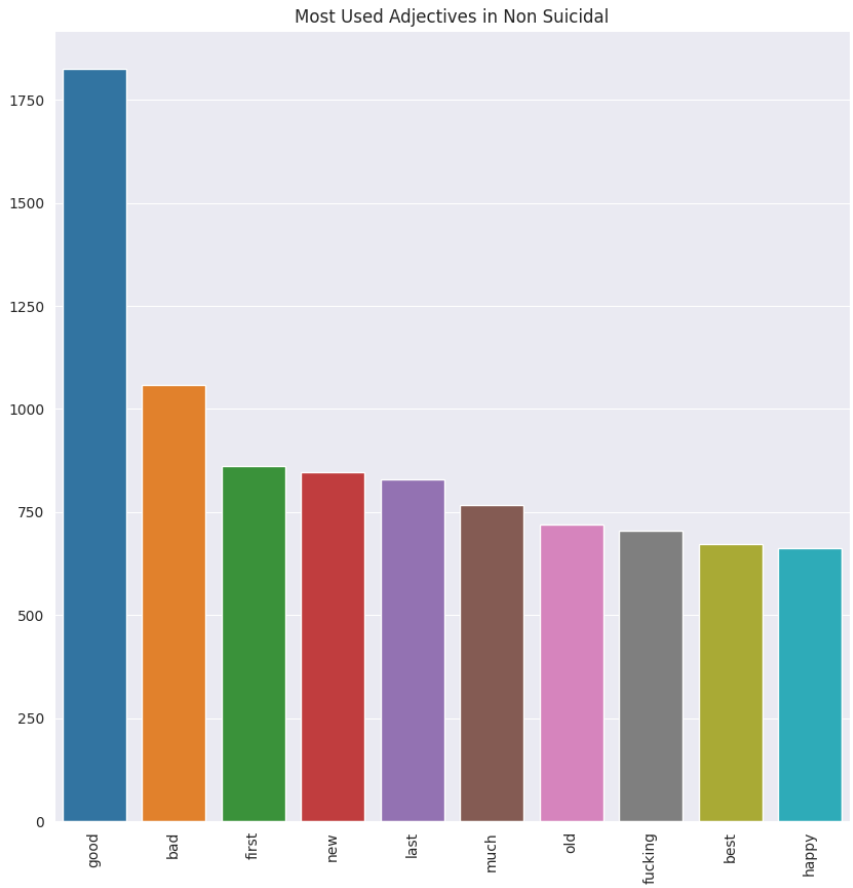


Figure: Most Used Adjectives on Reddits classified as non-suicide

BoW

We used Count Vectorize, and we ended up with a matrix with the number of lines equal with the number of reddits, and a number of columns equal with the number of unique words in the vocabulary (

- We shuffled the data
- We split the data as follows
  - 75 % for training
  - 25 % for testing

BoW

To implement BoW we followed 3 steps:

1. Preprocessed the data with Sci-kit learn
2. Build an internal vocabulary with fit method
3. Transformed the vector into an array, in order to create a sparse matrix.

SVM

For the SVM implementation we have used several regularization coefficients.

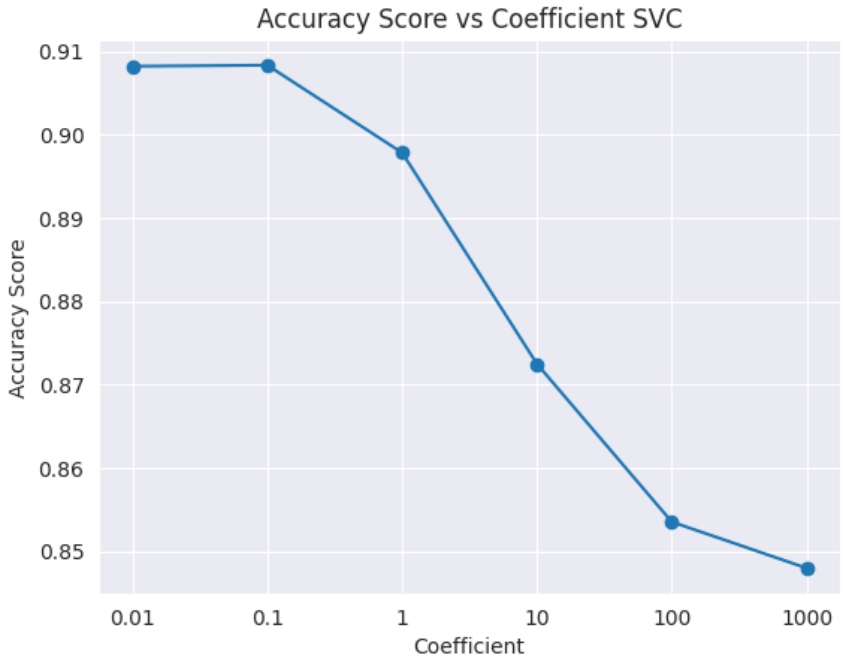


Figure: AccuracySVC

We concluded that the smaller the coefficient, the higher the accuracy is 0.92

SVM

Confusion Matrix

On the first try we have received Convergence Warning, so we increased the number of iterations, by 10.000, and we only tested for small coefficients.

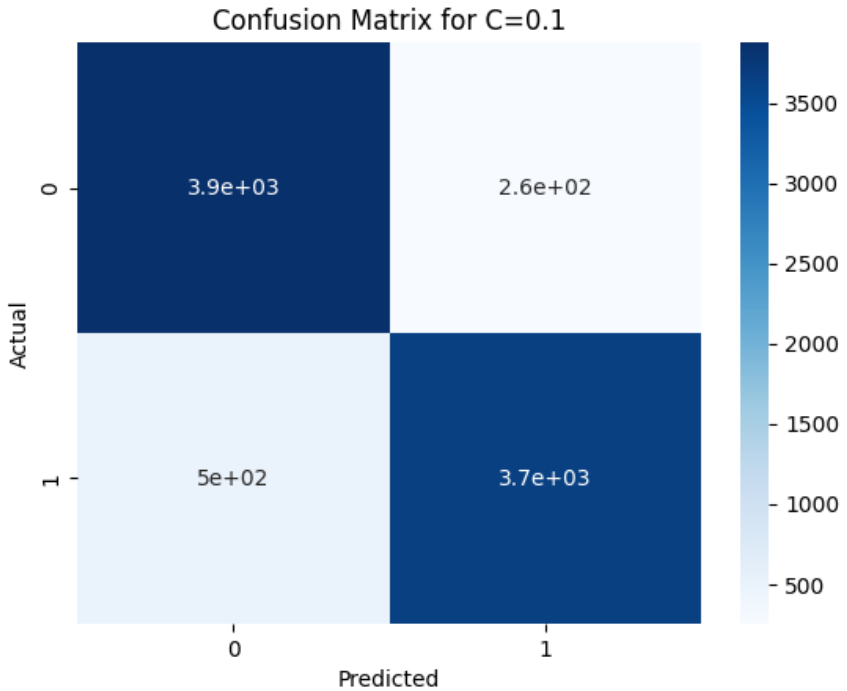


Figure: Confusion Matrix

NN first try

Initially, we have set an embedding dimension of 16, and used tf.Keras.seqluential as we wanted to learn our neural network epoch by epoch. For the layers we had:

- tf.keras.layers.GlobalAveragePooling1D()
- tf.keras.layers.Dense(24, activation = 'relu')
- tf.keras.layers.Dense(1, activation = 'sigmoid')

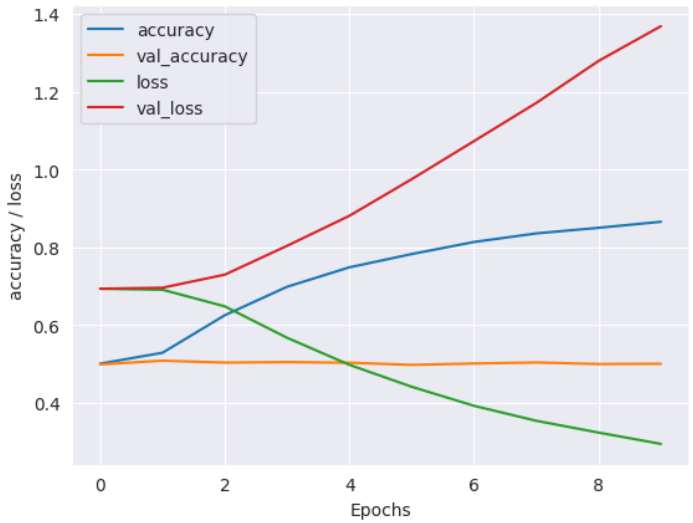


Figure: Training on 10 epochs

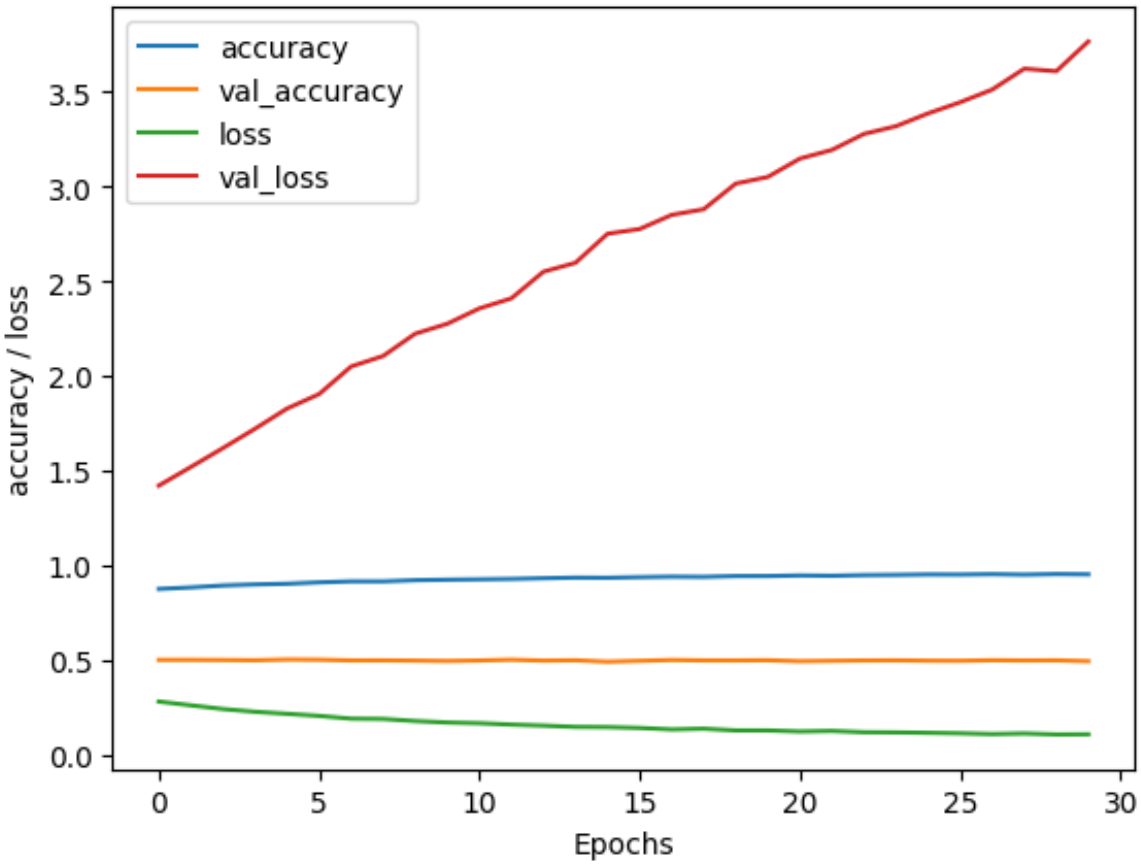


Figure: Training on 10 epochs

BERT

For the BERT model we used BERAT-base-uncased pre-trained model and tokenizer. We converted the inputs using dictionaries.
To train the BERT model we have used the exact same values as we used on NN, but the results was quite unimpressive.
The training was made on 30 epoch, because trying more it will take us a lot of time.

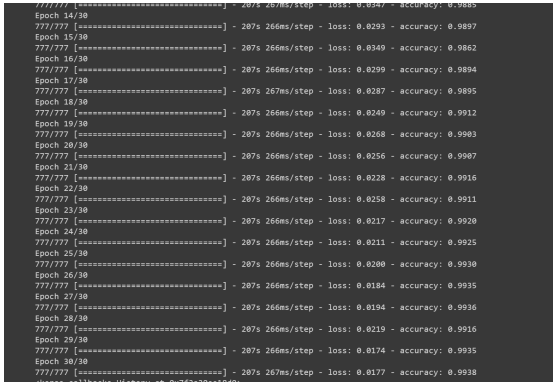


Figure: Training on 10 epochs

In the end, the total accuracy for the testing data was 0.50 .

Comparison

Model_Method	Accuracy
SVM	0.90
SVC	0.84
Gaussian Naive Baye	0.88
Neural Network	0.92
Bert	0.50

Concluzii finale

- The highest accuracy was obtained with Neural Networks.

Future work

- Wait to see if we are going to receive a data set we asked for
- Train a BERT model better, so we can get better results
- Try the experiment other languages