MED 277 Project Documentations

Jiaqi He, Shilu Jiang, Shulin Cao

Introduction

Classifiers and text mining are important and effective tools to aid analysis. There are plenty of cases where physicians need to diagnose diseases in order to provide patients with detailed and precise health information and further medical prescriptions. The process of translating this action into the language of computer science is a classification problem.

Mental illness is, more often, implicit and difficult to detect compared to physical diseases, which is consequently harder to get diagnosed and treated accordingly. Increasing number of people are now suffering from mental illnesses, a fraction of which will do harm to themselves and even commit suicide due to illness according to many reports.

Nowadays, many people are caught in mental depression due to various pressure. And it becomes more common that people involved in these mental diseases might choose to commit a suicide and lead to deep sorrow among their families and friends. Despite that a lot of efforts has been put in changing this condition,the results are apparently not so effective. A potential reason for this problem is that most people tends to hide their deep thoughts regarding their diseases to others due to social pressure or so until they cannot stand anymore and finally commit a suicide. What's worse, even if they hope to seek for help from doctors or psychological counselors, the complicated procedures and expensive costs on the medical treatments are another problem. The treatment resources are limited so the organizations are trying to select the real-targeted patients using lengthy survey while the survey takers may lie or fail to answer the question accurately subjected to limited time and energy. Thus, we tend to find out a way to detect the potential suicide population according to their experiences or words, which could be extracted from the social media, such as twitter, facebook and instagram which will be anonymous to others. Therefore, the medical organizations could come to them and offer help. Moreover, the pain of physical illness may cause patients depressed and tend to commit a suicide. Thus, by automatically evaluating their tendencies of commiting a suicide according to the therapeutica records, doctors and nurses can stop the patient from hurting themselves and do something to ease the mental stress.

Problem Definition and Significance

The dataset we used for the 2018 shared tasks comes from the National Child Development Study which is composed of short essays on where they saw themselves at age 25 written by children at age 11. It also include the prescription, anxiety and depression levels labeled by psychologists. In this project, we will mainly focus on mental health problems. Our goal is to analyze some related medical records and assess the patients' level of suicidal tendencies. More specifically, we utilize a collection of medical transcription records in the specific domain of Psychiatry/Psychology and expect to distinguish patients who are more likely to commit suicide from others. We want to find key features, similarities and differences among medical records by implementing some NER algorithms to help extract crucial named entities from plain texts and transforming text into vectors. Then we will classify medical records and find out targeted patients with the highest probabilities of committing suicides.

Methods

NLP preprocessing

Since the text data we used are the essays of children at 11 and due to the privacy policies, those texts requires a lot of preprocessing. First, inspect one of the examples. The essay would look like the following:

> My name is (name) and I am 25 years old. I am not married and I have a little boy. My life I am leading is not so good just now. My interests are gardening and jig-saw making I mak*e a jig-saw when I have notheng to do. My home life is not very intressting I just sit and watch the tv on Monday I watch Blue Petter. My work is an Builder and I have to build a 9 story flat, a factor, a school and for all thess job`s and a lot of outher`s. I need a scaffolding to stand on to get to the top of the building. I have to go up and down the scalffolding to get some bricks and cement. I have sore legs going up and down the scalffolding. By this time I have got good exersice . When I go home I am as fit as a fiddle for my dinner. I come back after my dinner up the scaffolding for more buillding and exersice. I go hom with my week's wages on my poket and I have got 10xxxx 15s and 6d for the week. My little boy (name) is staying with my next door neabour when I am at work . He has a train set their and it works by battery it only cost me 39. and 11 for it. My Garden is quite nice and I have a prize wining melling . I ceep it in a green house. When my favouret tv program comes on witch is How I go and watch it . It shown you how to left 8 brick with one hand last week. I do my Garden in the wonteh when the ground is soft. I *out my potatoes in the ***** and cover them ap so the frost dose not get them. After that I plant the onion's and shalots Words: 318.

We can clearly see some problems of these essays: (1) redundant words in “()/[]” (2) masked numbers (3) incorrect spellings. Hence, we need to preprocess the data before actually using them.

First step we did is to remove “()/[]”, then we remove punctuations stop words. Also, numbers-related words can be removed because numbers themselves don’t convey much information about the texts. We also took care of the word counts at the end of each essay. The pre-processed essay looks like the following:

> name years old married little boy life leading good interests gardening jigsaw making make jigsaw notheng home life intressting sit watch tv monday watch blue petter work builder build story flat factor school thess jobs lot outhers need scaffolding stand get top building go scalffolding get bricks cement sore legs going scalffolding time got good exersice go home fit fiddle dinner come back dinner scaffolding buillding exersice go hom weeks wages poket got week little boy name staying next door neabour work train set works battery cost garden quite nice prize wining melling ceep green house favouret tv program comes witch go watch shown left brick one hand last week garden wonteh ground soft potatoes cover ap frost dose get plant onions shalots

Last step is to tackle incorrect spellings in the essay.

We use a simple spelling corrector model provided by Peter Norvig. The spelling corrector is based on Baye’s theorem, as the following equation shows:

$$\text{argmax}_{c \in candidates} \, P(c) \, P(w|c) \, / \, P(w)$$

We selects words within 2 edit distance away from the target word w as our candidates c and then we chose the word with highest probability as the result of spelling correction. The word set is extracted from three sources: Project Gutenberg, Wiktionary and British National Corpus.

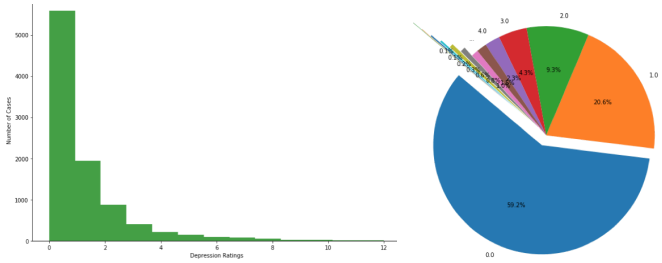
The sample results of spelling corrector is :

```
print(correct('korrect'))
print(correct('corrcet'))
```

```
correct
correct
```

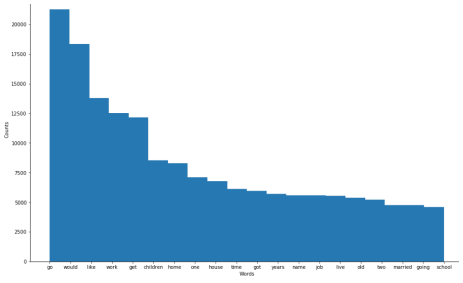
After the preprocessing of the datasets, we have performed some basic analysis on the structure of the datasets. Preliminary results on the dataset shows that the data labeled 0 makes up around 60% of the whole dataset. Our preprocessing methods convert these data to ones that are only labeled 0 and 1.

Labels in range between 0 and 12 with distribution showing in the figure
In the project, we will label all data essay with two groups:
0: all depression rating 0
1: all other depression ratings



Bag of Words

Then using the bag of words algorithms, we proposed two different methods to present the frequency of a word. One is based on the counts of the different words in the whole datasets. We ranked the words based on its appearance times in the whole dataset and propose a corresponding vector which then applied to each individual document to calculate each word's frequency. Top 20 frequently words shown as following picture.



TF-IDF

We have also used the tf-idf method to calculate the frequency of each word in each dataset.

"Term frequency": $tf(t,d)$ = number of times the term *t* appears in the document *d*

e.g. tf("Taylor Swift", that news article) = 3

"Inverse document frequency": $idf(t,D) = log\frac{N}{|\{d \in D : t \in d\}|}$

term (e.g. "Taylor Swift") set of documents

"Justification": $P(t|D) = \frac{|\{d \in D : t \in d\}|}{N}$ so $idf(t,D) = -\log P(t|D)$

Word Embedding-word2vec

Word2Vec is a two-layer neural network model commonly used to create word embeddings. It takes as input a large corpus of text and produces a vector space. In this project, we utilized the pre-trained Google's Word2Vec models to transform each essay into a vector of 300 dimensions.

Apart from this word embedding, we also included two statistical features that we acquired during the preprocessing part: one is the ratio *r* of words that are spelled incorrectly (defined below), the other feature is the total number of words, represented as *T*, in the processed essays.

$$r = \#\ of\ incorrectly\ spelled\ words\ /\ T$$

$$T = \#\ of\ words\ left\ in\ the\ processed\ essay$$

For better outcomes, we normalized the second feature *T*, the total number of words in the processed essays, to a train of data with zero mean and unit standard deviation.

Therefore, the structure of input would be in the form of a 9455×302 matrix, shown as below.

For the goal of classification, we tried three different models: random forest classifier, Adaboost classifier, and neural network model. We split the data into training dataset and test dataset by the ratio of 0.2, namely, we selected 80% of the data to be training data and the remaining 20% would be the test data.

Results

Bag of Words

We used the count method together with logistic regression model to train the dataset. With an increase of word vector length, our training accuracy is keep improving. The test accuracy, however, goes up first and then goes down. The reasons behind this special phenomenon are mainly focused on the potential over-represented and the imbalance of label distribution from the dataset. Increasing size of word vector would help trained the data to be more fitted to the labels and also help improve the performance of the original model. But after reaching a bottle neck point, this will not be able to increase the test accuracy anymore since the testing dataset

size is limited, and consequently affect its performance in the test dataset. Also, we noticed that the imbalance dataset will make the training more biased through this process.

Accuracy\Words	20	200	2000
Training	58.87%	63.91%	78.13%
Test	58.25%	60.56%	57.10%

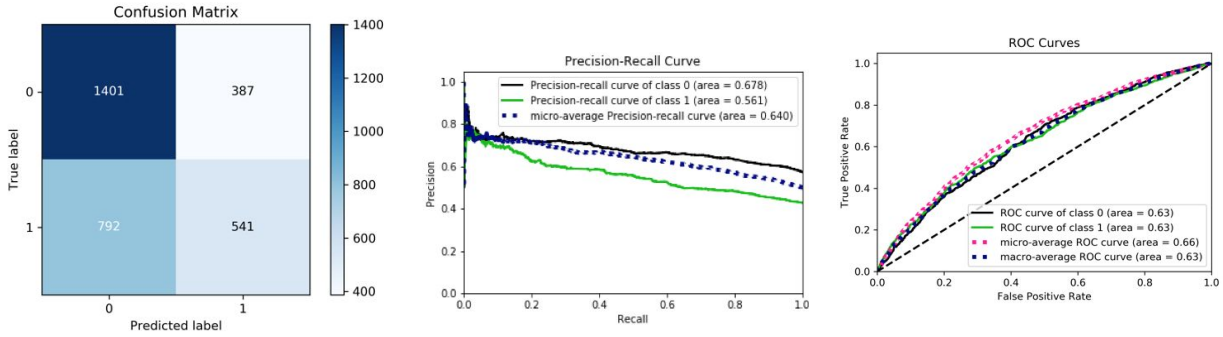
Also, we have conducted the support vector classifier together with the count method. This new method brings a better test accuracy compared to the previous model.

Recall R = TP/(TP+FN) = 0.41

Precision P = TP/(TP+FP) = 0.58

F1 score = 2*R*P/(R+P) = 0.48

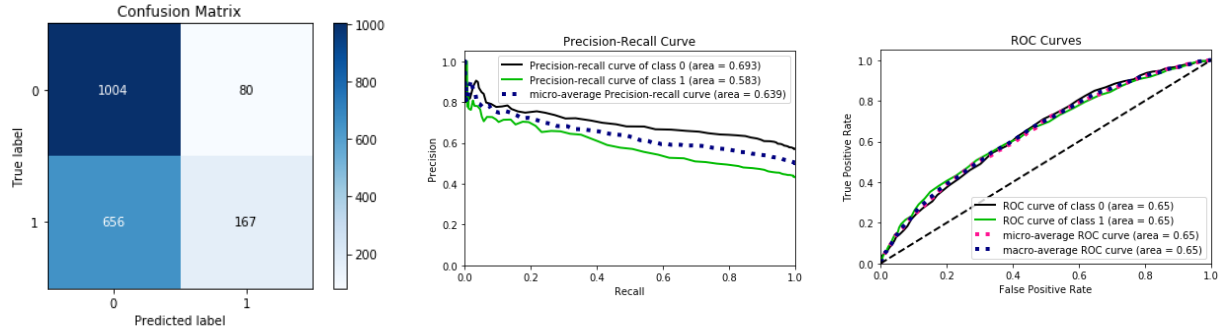
accuracy: 62.223646 %



(a) Confusion matrix (b) Precision-Recall curve (c) ROC curve

TF-IDF

We use the most simple machine learning algorithm KNN(K-Nearest Neighbors Algorithm) based on tf-idf matrix, and attains the following results on test set after training.



(a) Confusion matrix (b) Precision-Recall curve (c) ROC curve

KNN classifier accuracy: 61.405349 %

Recall R = TP/(TP+FN) = 0.20

Precision P = TP/(TP+FP) = 0.68

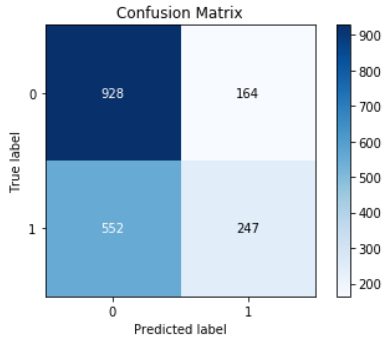
F1 score = 2*R*P/(R+P) = 0.31

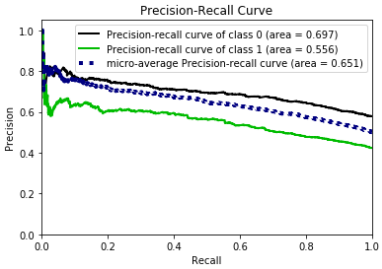
Compared with the previous model, the ROC is slightly better but Recall and F1 score is fall less than before. That's probably because the knn model simply use 0.5 as the threshold to classify 0 and 1.

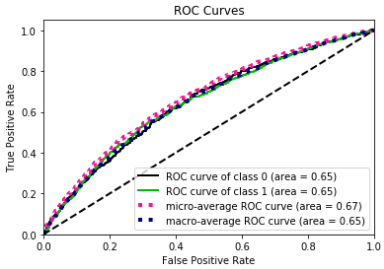
Word2Vec embedding

(1) Random forest classifier

In this model, we set n_estimators=1000, max_depth=15, verbose=1, class_weight='balanced'. The accuracy on the training dataset reached 0.992, while the accuray on the test set is 0.621. The confusion matrix, PR curve and ROC curve are shown below.



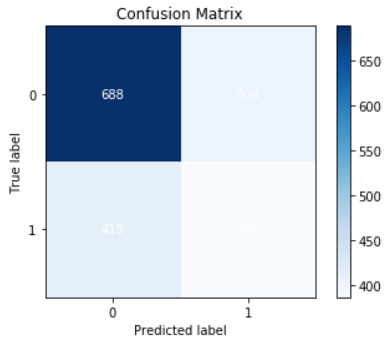


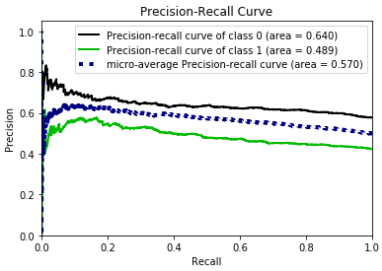


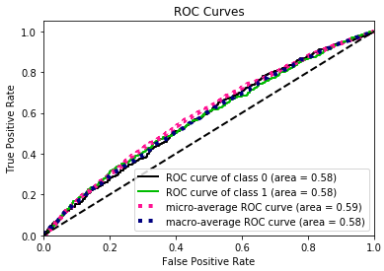
(a) Confusion matrix (b) Precision-Recall curve (c) ROC curve

(2) Adaboost classifier

In this model, we set DecisionTreeClassifier(max_depth=2, class_weight='balanced'), algorithm="SAMME.R", n_estimators=1000. The accuracy on the training dataset reached 1.000, while the accuray on the test set is 0.567. The confusion matrix, PR curve and ROC curve are shown below.



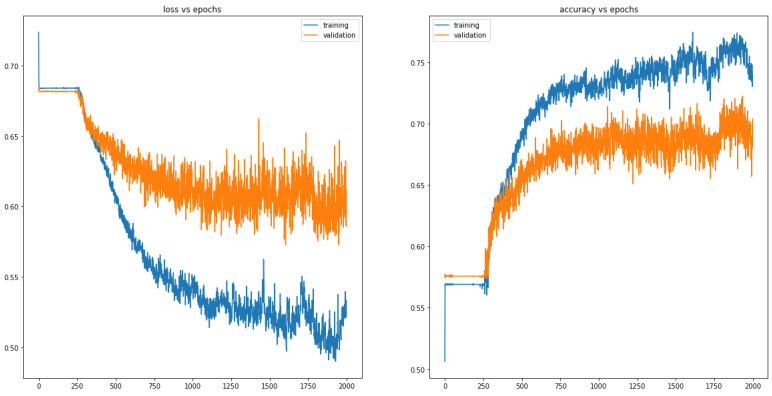




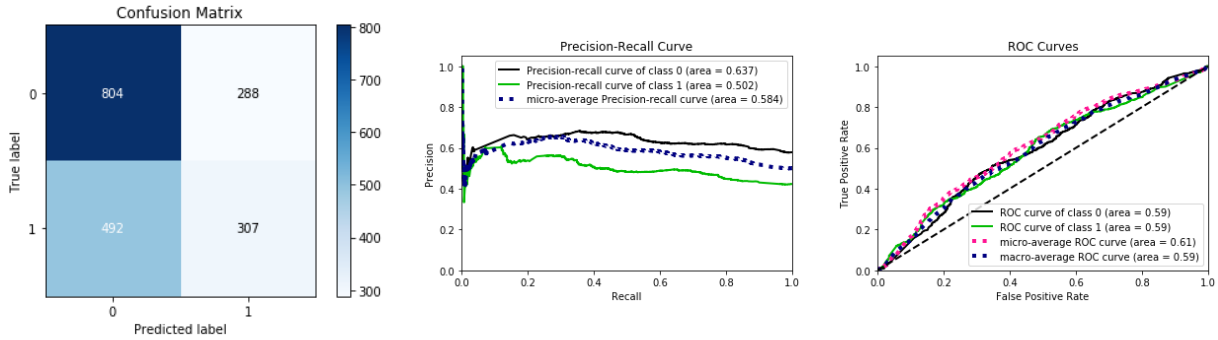
(a) Confusion matrix (b) Precision-Recall curve (c) ROC curve

(3) Neural network model

In this model, we created a 3-layer neural network with hidden layer sizes set to be 60, 24 and 5 respectively. Moreover, we set a drop-out rate of 0.5 on the first and the second layer. For training, we set the training epoch to be 2000, and in each epoch, we take out 20% of the training data to be the validation set so as to monitor the model performance.
The loss and accuracy on the training and validation set is shown below.



We can see from this figure that after 2000 epochs, the accuracy on the training and validation set has become steady and vibrates within a small bound. The final accuracy on the training set is 0.763 while the accuracy on the test set is 0.587. The confusion matrix, PR curve and ROC curve are shown below.



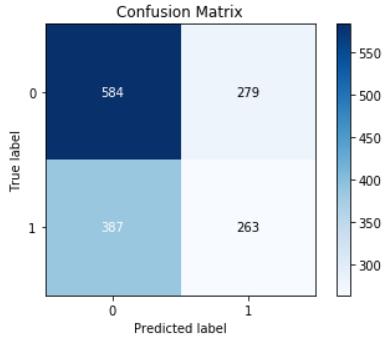
(a) Confusion matrix (b) Precision-Recall curve (c) ROC curve

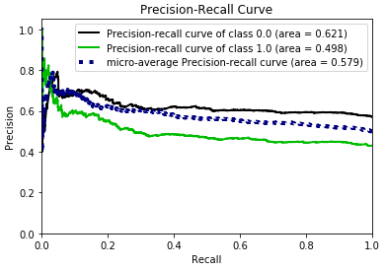
LSTM model

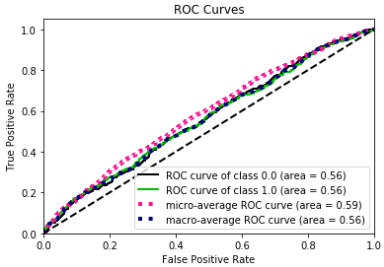
Long short-term memory (LSTM) units (or blocks) are an important component for layers of a RNN. It is unique for having a cell, an input gate, an output gate and a forget gate. We established our LSTM model in the following format:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 2000, 128)         256000
_________________________________________________________________
lstm_3 (LSTM)                (None, 196)               254800
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 394
=================================================================
Total params: 511,194
Trainable params: 511,194
Non-trainable params: 0
```

We then train this model and evaluate this model on the test set. The accuracy on the training set is 0.559 while the accuracy on the test set is 0.646. The confusion matrix, PR curve and ROC curve are shown below.







(a) Confusion matrix (b) Precision-Recall curve (c) ROC curve

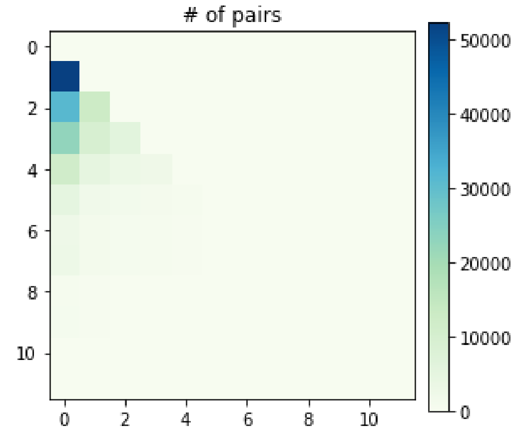
Relabelling Data

In spite of the various models we tried to classify the data, the performances of these models are not ideal. Generally speaking, we have approximately 0.60 accuracy on the test set. Also, from these confusion matrices, we can see that the recall is relatively low.

From the figure of training accuracy in neural network model, we can see that the training and validation accuracy stops to become better at around 0.75 and 0.65 respectively. The phenomenon of low training accuracy implies that there may not be strong connections between the input data and the expected labels.

If we inspect the input data in terms of the pairwise cosine similarity, and count the numbers of cases where two entries with different labels have high similarity (cosine similarity > 0.95), we can get the following matrix. The number N_{ij} located at row *i* and column *j* in the matrix means that there are N_{ij} pairs of data, one has “depression” level *i* and the other has “depression” level *j*, and yet they have cosine similarity greater than 0.95. The figure below demonstrates the distribution of this matrix in a manner of countings.

0	0	0	0	0	0	0	0	0	0
52472	0	0	0	0	0	0	0	0	0
31297	12631	0	0	0	0	0	0	0	0
22927	9546	5873	0	0	0	0	0	0	0
11511	4830	2929	2274	0	0	0	0	0	0
5004	2019	1243	941	487	0	0	0	0	0
2659	1117	697	541	289	121	0	0	0	0
2786	1201	730	569	292	115	70	0	0	0
521	216	131	97	48	22	13	13	0	0
633	268	159	120	63	25	16	17	3	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



(a) Counting matrix (b) color representation of the counting matrix

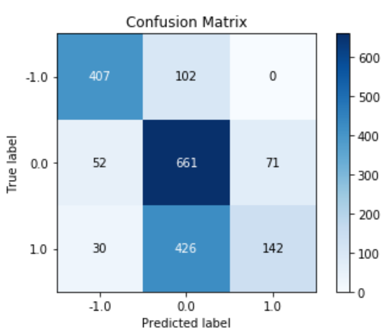
What's astonishing is that there are many pairs of data that satisfy the conditions we have set. This implies that there are many data entries where they are highly similar to each other from the perspective of these essays but the writers of these essays actually experienced different levels of depression. In other words, these essays were from people at different depression levels and yet they look almost the same!

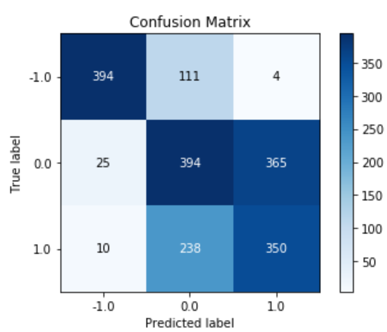
Thus, simply using these essays as inputs, we actually cannot distinguish them from each other. Therefore, for the sake of better performance, we might as well put aside these highly similar data entries which are from people at different depression levels.

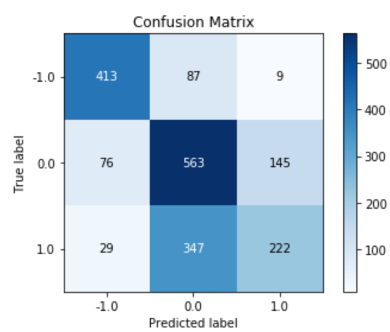
If we pick out data entries involved in this particular case, we found 2694 such entries. We label them -1 to indicate that we are currently lack of effective and tenable information to make classifications on them and therefore tentatively mark them -1 to indicate this fact.

After relabelling, we re-did the modeling and get the accuracies and confusion matrices below for the three models using Word2Vec embeddings.

From the results, we can notice that the performance on the test set has improved by roughly 5%. Also, the confusion matrix is distributed in a more reasonable way.







Test accuracy = 0.640 Test accuracy = 0.602 Test accuracy = 0.634

(a) Random forest (b) Adaboost (c) Neural network

Conclusions and discussion

1.Our model performance can achieve 65% accuracy, and from confusion matrix we can see s high true negative rate but a relatively low true positive rate.

a.Our dataset shows 59% labeled 0 and others labeled more than 1 (NOT WELL BALANCED)

b.This dataset from age 11 children who may have limited vocabulary, which limit our choice of word features

c.Unable to represent the true meanings from them though correcting the words and spelling (LOSS)

d.Preprocessing for essays still leaves some useless words which may have higher frequent appearance.

2.Probably the correlations between the essays and their depression degrees may not be highly related. So in the future we hope that we can get better features so that we can perform better models for children's mental health.

Related work:

1. Loveys, Kate, et al. "Small but Mighty: Affective Micropatterns for Quantifying Mental Health from Social Media Language." Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology—From Linguistic Signal to Clinical Reality. 2017.
2. Jamil, Zunaira. Monitoring Tweets for Depression to Detect At-risk Users. Diss. Université d'Ottawa/University of Ottawa, 2017.
3. Guo, Jia-Wen, et al. "A Corpus Analysis of Social Connections and Social Isolation in Adolescents Suffering from Depressive Disorders." Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology—From Linguistic Signal to Clinical Reality. 2017.
4. Bullard, Joseph, et al. "Towards Early Dementia Detection: Fusing Linguistic and Non-Linguistic Clinical Data." Proceedings of the Third Workshop on Computational Lingusitics and Clinical Psychology. 2016.
5. Desmet, Bart, Gilles Jacobs, and Véronique Hoste. "Mental Distress Detection and Triage in Forum Posts: The LT3 CLPsych 2016 Shared Task System." Proceedings of the Third Workshop on Computational Lingusitics and Clinical Psychology. 2016.
6. Franco-Penya, Hector-Hugo, and Liliana Mamani Sanchez. "Text-based experiments for predicting mental health emergencies in online web forum posts." Proceedings of the Third Workshop on Computational Lingusitics and Clinical Psychology. 2016.