Stress Anxiety Monitor (SAM)

New Mexico
Supercomputing Challenge
Final Report
April 4, 2023

Team 27
Monte del Sol Charter School

Team Members:
Gabriella Armijo
Emlee Taylor-Bowlin
Teacher Sponsor:
Rhonda Crespo
Project Mentor:
Mark Galassi

Contents

1 Executive Summary

We have all lost someone prematurely due to a tragic suicide. It often leaves us with a feeling of dread and many "what if"s. What if I saw the signs, what if I had stopped them, what if I had texted them that night? This isn't a burden you should have to bear. Imagine a simple program that could read some text snippets and alert you if you or a friend needed to reach out to a professional? In this model we used machine learning to create an AI that will do just that.

SAM (Stress Anxiety Monitor) is an AI built off of the Random Forest Classifier model. It is trained with data from the "dreaddit" data set (Turcan & McKeown, 2019).

SAM uses two different training and processing approaches: the Bag of Words and Intensity Value methods. The Bag of Words method takes a count of every word in the text snippet. It then feeds this data into a sparse vector, and any word in the English dictionary that it does not find becomes a zero. The words are then processed by frequency, so redundant words like the, as, and a are not counted as heavily. The Intensity Score method makes a weighted score of the "worrisome" key words that show up in the text.

This word data is then fed into a machine learning Random Forest Classifier that processes the data and makes accurate predictions about a person's risk. We found SAM to be accurate 99% of the time on the in-sample set. SAM is 93% accurate on out of sample data. Hyperparameter tuning is still needed for SAM to be as accurate as possible.

As of now SAM can be used to help friends and family recommend people to professional help. We would like to turn SAM into a simple webpage that could process screenshots and take text input. SAM can help reduce the total number of teenage suicides in New Mexico, and keep our communities safer and happier.

2 Motivation and Plan of the Work

2.1 Suicide

Suicide is the #2 leading cause of death among teenagers and young adults worldwide (D'Hotman et al., 2020; "Suicide statistics and facts. SAVE.", 2021). New Mexico is currently ranked #4 in the nation for states with the highest suicide rates. ("America's Health Rankings." 2020) New Mexico's current teen suicide rate is 24 deaths per 100,000 adolescents ages 15-19 a year ("Suicide rates by state 2023." 2023). This results in about 513 teen suicides a year.

Suicide is preventable with intervention; however, it is hard to intervene when people don't openly talk about their feelings and it is easy to miss the warning signs. In this day and age many teens and young adults spend a majority of their time chatting on social media or via text messaging. Cries for help may be easily masked in this dull form of communication. Being dependent on technology means that many teens are afraid of reaching out and talking to someone in person for help, meaning the only way they can convey their emotions is through texts and many hope that someone will pick up on their cry for help.

With the help of machine learning, however, suicide prevention may be easier. We created an AI named SAM that is able to go through social media posts and/or text messages looking for trigger words that may indicate that a person may be showing signs of high stress that may lead to suicide. We hope SAM will lead to early detection and prevention of suicide.

2.2 Project Goal

The goal of this project is to accurately determine if a person may be at high risk of suicide or self-harm based on messages that people share online. Our hope is that SAM can be applied to help notice warning signs early on in things as simple as a text message.

3 Model and Methods

3.1 Obtaining and Preparing Data

Our data set is publicly available on the Kaggle web site:

https://www.kaggle.com/datasets/adtysregita/stress?select=stress.csv

This data set initially contained 2,343 rows and 116 columns in a csv file. All 2,343 rows were usable and didn't contain any missing values. We only use 4 of the columns: the row number, the text, the label, and the confidence.

text	id	label	confidence
He said he had not felt that way before, sugget	33181	1	0.8
Hey there r/assistance, Not sure if this is the ri	2606	0	1
My mom then hit me with the newspaper and it	38816	1	0.8
until i met my new boyfriend, he is amazing, he	239	1	0.6
October is Domestic Violence Awareness Mont	1421	1	0.8

Figure 1: Snippet of data from worrisome messages.

The label is a binary label that codes for stress or no stress. A label of "1" indicates stress, whereas "0" does not. These numbers were provided by professionals based on the text for that row. Confidence is another professional-provided value that indicates the confidence the expert holds in that label. At this time SAM does not use the confidence metric.

This data is then finally ready to feed into the Random Forest Classifier that we wrote for this project, using the *scikit learn* framework. (Amadebai, 2022; Kessler et al., 2016)

3.2 An Introduction to Random Forest Models

Random Forest Classifiers and Random Forest Regression are both methods of machine learning. Random Forest models take numeric values and feed them into decision trees. The trees then decide where this data continues and then break into two or more trees. They inevitably end up as decision leaves. These leaves are the final output of the decision and are also numeric values. In the case of regressors, they can be a range, why classifiers give a binary value.

Figure 2 shows an example of a Random Forest Regressor. It allows for many different outputs like apple, cherry, or strawberry. It makes it's decisions based off of size.

Figure 2: Decision based off a Random Forest Regressor model. Image borrowed from:
https://developer.nvidia.com/blog/accelerating-random-forests-up-to-45x-using-cuml/

The following figure is an example of a Random Forest Classifier. It can only provide a binary output, in this case yes or no. Our model uses a classifier to give the output of stress or no stress.

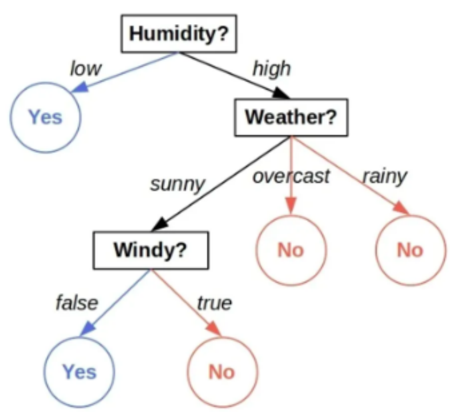


Figure 3: Example of random forest classifier.

3.3 Implementation of a Random Forest Classifier

We used a Random forest Classifier as our model type. It produces a 1 to indicate stress or a 0 to indicate no stress. Many programmers try different model types: we deduced that Random Forest Classifier by far is the most accepted model for natural language learning. We manipulated two different ways of processing the data, and before it was fed into the model.

3.4 Intensity Score Method

The first method of data processing was the Intensity Score method. We wrote a dictionary of “worrisome” keywords, which are words the literature suggests can indicate stress. Each word was given a weight, and the weighed word occurrences are added.

The result was a training set consisting of a single numeric score for each text snippet, and the “expert label” of stress or no stress for that snippet.

This method makes for a very simple initial implementation, but it lacks nuance: the specific use of word pairs is lost in that single score.

3.5 Bag of Words Method

The Bag of Words (Brownlee, 2019) is far more inclusive and more complex. The Bag of Words method reads through the text snippet and accounts for the total number of times any word from the English dictionary appears. It then scales the numbers by importance. Importance is decided by the number of times each word occurs, so words like the, as, and, then will carry far less importance, and not affect the interpretation.

This method is far more reliable, because it accounts for all possible words. However it does come with the downside of producing a sparse vector. A sparse vector is a vector that contains many zeros, showing no data. It can be useful if we see zeros in spots that would indicate words like suicide, depression, or lonesomeness. It does become a nuisance when zeros are always seen to indicate words like Pneumonoultramicroscopicsilicovolcanoconiosis (considered the longest word in English and is used for a certain lung disease).

4 Software Demonstration

Our code can be found at the following repo:
https://codeberg.org/EmleeTaylorBowlin/SAMonitor

Here is an example of how you could clone it and then run the training and analysis;

we include the output, from which you can see the error estimate and the score on both in-sample and out-of-sample data.

SAMonitor.py also produces plots and saves them to disk. You can see in Figures 4, 5, and 6.

Note that the program has a -help option if you want to see how to test it in more subtle ways.

```
git clone \url{https://codeberg.org/EmleeTaylorBowlin/SAMonitor.git}}
# Make sure you have the following libraries also installed on your computer
# pandas, numpy, matplotlib, and scikitlearn.
# Begin by running the programs in this order:
cd code/SAM
$ ./SAMonitor.py TnPData/dreaddit-train.csv TnPData/dreaddit-test.csv
Hey! SAMonitor here! We will now train on data and analyze snippets.
# loading training data from: TnPData/dreaddit-train.csv
# in-sample error metric (mae): 0.0014094432699083862
# in-sample classifier score: 0.9985905567300916
# saved confusion matrix to file confusion_insample_bag-of-words.png
# ========= out of sample =========
# out-of-sample error metric (mae): 0.3062937062937063
# out-of-sample classifier score: 0.6937062937062937

APPLY_TO_TEXT: worrisome score: """Things had started so well, but
now, two years later, I sit listening to Janacek's quartets - the
Intimate Letters and the Kreutzer Sonata - and I find myself feeling
desperate, and wondering if my life will ever be what I had dreamed it
would be.  Well-meaning people offer me words of wisdom, like 'it gets
better' and 'just navigate the terrain you are in' and 'do not feel
that you are below the standard everyone else is at'.  I know they are
```

```
right and I hope that some day I can take their advice, but I cannot
yet promise that I will.""" =====> [1] i.e. DO_WORRY

APPLY_TO_TEXT: worrisome score: """I was walking down the street,
singing do wa diddy diddy dum diddy do, and when I turned the corner I
saw someone walking with a book by Murakami, my favorite author of
Japanese magical realism.  We started talking and shared a lot of
interests.  This was a year and a half ago, and now we're together
almost every single day, singing do wah diddy diddy dum diddy do."""
=====> [0] i.e. NO_WORRY
```

5 Results

Our final model accurately predicts if someone is at high levels of stress based off of snippets of texts. We were able to test the applicability of our model by feeding it data sets of different sizes, and having our Random Forest Classifier print various performance metrics, namely the "mean absolute error" (MAE), the "accuracy score", and for visualization purposes the "confusion matrix".

Our discussion of metrics is based on the "dreaddit" training (2838 samples) and testing (714 samples) data sets.

A confusion matrix is one of the ways to express if the classifier's predictions were correct or incorrect. The confusion matrix generates both actual and predicted values. All the diagonal elements denote correctly classified outcomes. The misclassified outcomes are represented in the off-diagonal portions of the confusion matrix. They are commonly denoted as true positive, true negative, false positive, and false negative.

In our application the main concern is the lower left entry in the confusion matrix: when we predict "OK" but we should really have been worried. These are called "false negatives", and would lead SAM to ignore a call for help.

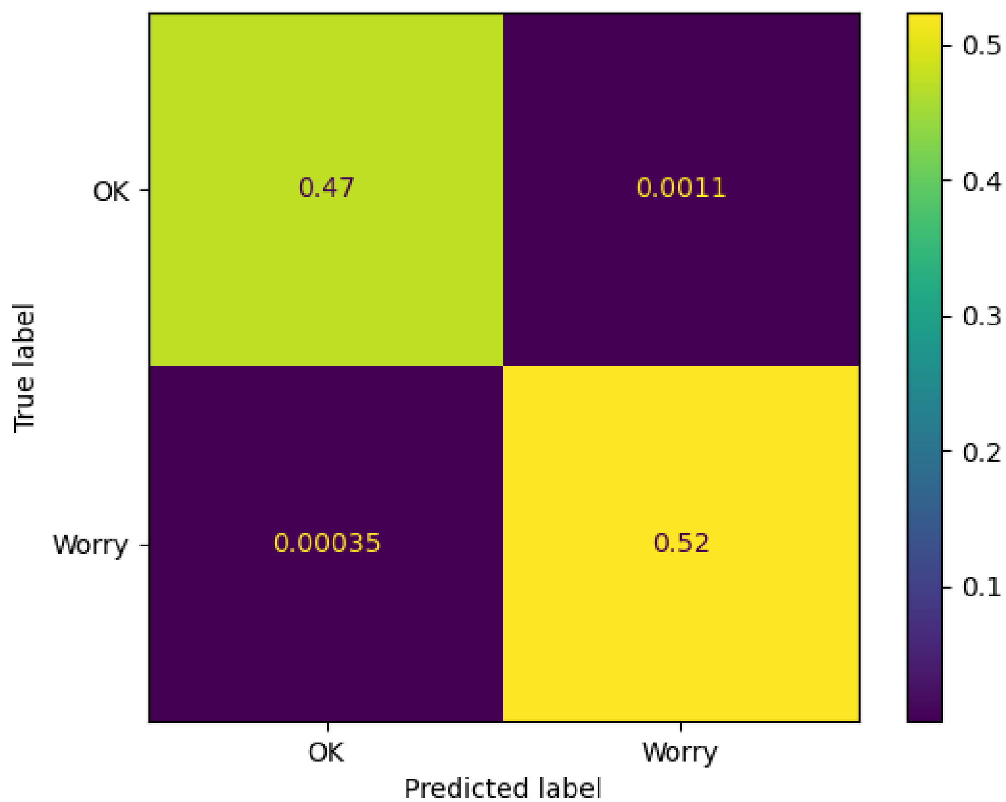
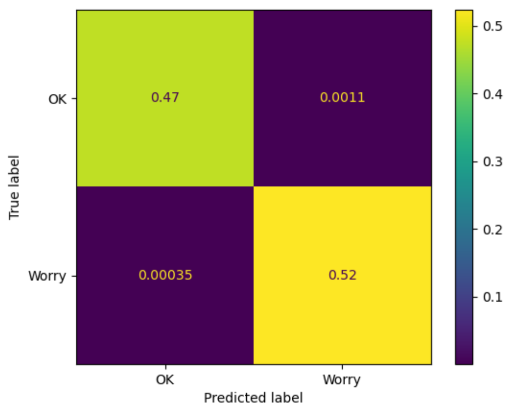


Figure 4: Simple example of confusion matrix for in-sample data on our training set.

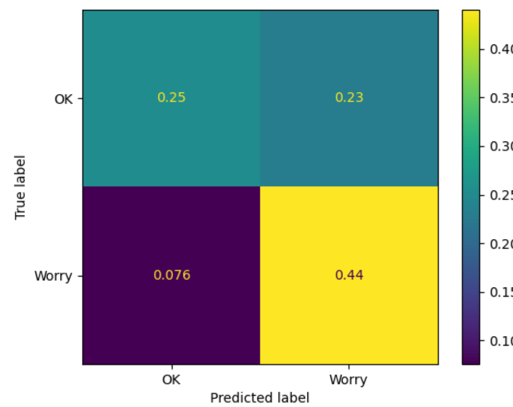
in-sample MAE	in-sample accuracy	out-of-sample MAE	out-of-sample accuracy
0.0014	0.99859	0.30629	0.6937

In the confusion matrix in Figure 4 the top left value represents the **True Negative** where the model correctly predicted that the person is stressed. The top right value is a **False Positive** that says that the AI incorrectly predicted that the person was stressed (when they were actually OK). The bottom left is the **False Negative**, the AI incorrectly predicted the person to be ok (when they are actually stressed – this value is the key to whether our method works). The bottom right is a **True Positive** where the model correctly predicted that the person is not stressed.

We can run our model with either the bag-of-words method or the intensity score method: the user can switch with a command-line option. We use the Bag of Words method by default. Our model compares its predictions to the actual results and produces the confusion matrices. For either method, our model trained itself on a user specified data set sample before it tested itself on data set sample that it has never seen before. Using both methods we were able to deduce that the Bag of Words method was more effective in correctly classifying if someone was stressed or not.



(a) Training Predictions



(b) Testing Predictions

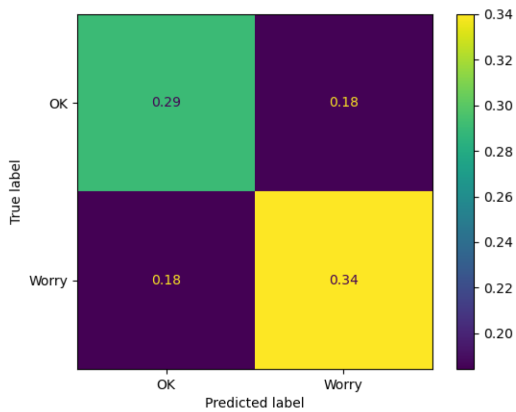
Figure 5: Bag of words predictions: here we look at the confusion matrix for both in-sample and out-of-sample data with the bag-of-words approach.

Figure 5 shows the confusion matrix for the Bag of Words method. In this instance the "in-sample" comes from the dreaddit training data, and the "out of sample" comes from the dreaddit test data.

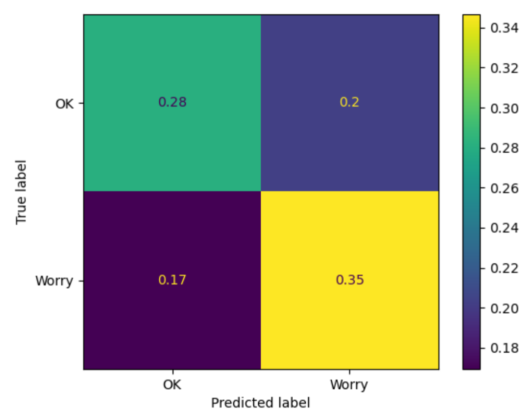
In Figure 6 we see what we get with the "intensity score" approach, used on the same data sets.

The main take-home from these figures comes from the lower-left square in the confusion matrix. The in-sample "false negative" rate with the bag-of-words approach is extremely low (0.00035, or 0.035%). With the intensity score approach it is much bigger (0.18, or 18%).

The method performs worse with out-of-sample data: 0.076 (or 7.6%) false negatives for bag-of-words, versus 0.17 (17%) false negatives for the intensity score approach.



(a) Training Predictions



(b) Testing Predictions

Figure 6: Intensity score approach: confusion matrix for in-sample and out-of-sample data.

Clearly we can now focus on using the bag-of-words approach.

6 Conclusion

6.1 Takeaways

In conclusion, we were able to make a functional AI stress monitor that was able to detect if someone was considered to be at a high stress risk. This could potentially be used in the future to aid health care officials in screening someone as high risk for suicide based on text messages. It could also help friends and family monitor loved ones and guide them to professionals. Through two ways of extracting data we where able to identify that a Bag of Words approach was the best option when it came to accurately preparing the data predict if someone was considered highly stressed or not. The largest takeaway of SAM is its ability to help friends and professionals intervene before suicide occurs.

6.2 Limitations and Errors

Our current out-of-sample performance is that SAM will incorrectly think that someone “is fine” about 7% of the time. How can this be improved?

One enhancement would be to include other emotions conveyed through the text, like sorrow, aggression, shock, etc. Many of these scores are available in the data set. By including more parameters in determining if someone is more at risk for committing suicide will make our model more accurate and therefore more useful to health professionals.

Hyperparameter tuning is also needed; this would entail changing the hyperparemeters. Hyperparameters are values that are used to control the learning process of the AI.

6.3 Future Work

Given time we would like to expand this project to include other emotions conveyed through text. We would also like to have SAM keep a running tab on how often a certain user shows high risk signs through multiple posts and the number of times they post. We would also like to add a web-based application that can take images and screen shots, as well as text. It could also have a user type answers to questions about how they are feeling. The text would then be fed to SAM for classifcation of stress level.

7 Acknowledgments

We would like to thank our teacher sponsor, Rhonda Crespo, for providing us with support needed to work on and finish this project on time, our mentor, Mark Galassi, for his expertise and guidance with the code involved in our project, and the Supercomputing Challenge, for their resources and support.

8 Further reading

1. Accelerating random forests up to 45x using cuml. NVIDIA Technical Blog. (2022, August 21). Retrieved April 4, 2023, from https://developer.nvidia.com/blog/accelerating-random-forest

2. Ahr. America's Health Rankings. (n.d.). Retrieved from https://www.americashealthrankings.org/explore/annual/measure/Suicide/population/suicide_15-24/state/NM

3. Amadebai, E. (2022, August 12). Decision trees vs. random forests – what's the difference? Analytics for Decisions. Retrieved from https://www.analyticsfordecisions.com/decision-trees-vs-random-forests/

4. D'Hotman, D., Loh, E., & Savulescu, J. (2021, June 1). Ai-enabled suicide prediction tools: Ethical considerations for medical leaders. BMJ Leader. Retrieved from https://bmjleader.bmj.com/content/5/2/102

5. Haque, M. U., Dharmadasa, I., Sworna, Z. T., Rajapakse, R. N., & Ahmad, H. (2022, December 12). "I think this is the most disruptive technology": Exploring Sentiments of ChatGPT Early Adopters using Twitter Data. Retrieved from http://export.arxiv.org/abs/2212.05856v1

6. Kessler, R. C., van Loo, H. M., Wardenaar, K. J., Bossarte, R. M., Brenner, L. A., Cai, T., Ebert, D. D., Hwang, I., Li, J., de Jonge, P., Nierenberg, A. A., Petukhova, M. V., Rosellini, A. J., Sampson, N. A., Schoevers, R. A., Wilcox, M. A., & Zaslavsky, A. M. (2016, January 5). Testing a machine-learning algorithm to predict the persistence and severity of major depressive disorder from baseline self-reports. Nature News. Retrieved from https://www.nature.com/articles/mp2015198

7. Nichol, A. (2022, July 1). Dall·E 2 pre-training mitigations. OpenAI. Retrieved January 6, 2023, from https://openai.com/blog/dall-e-2-pre-training-mitigations/

8. Reduce the suicide rate - MHMD-01. Reduce the suicide rate - MHMD-01 - Healthy People 2030. (n.d.). Retrieved from https://health.gov/healthypeople/objectives-and-data/browse-objectives/mental-health-and-mental-disorders/reduce-suicide-rate-mhmd-01

9. Walsh, C. G., Ribeiro, J. D., & Franklin, J. C. (n.d.). Predicting risk of suicide attempts over time through ... - sage journals. Predicting Risk of Suicide Attempts Over Time Through Machine Learning. Retrieved from https://journals.sagepub.com/doi/10.1177/2167702617691560

10. Suicide rates by state 2023. (n.d.). Retrieved from https://worldpopulationreview.com/state-rankings/suicide-rates-by-state

11. Suicide statistics and facts. SAVE. (n.d.). Retrieved from https://save.org/about-suicide/suicide-statistics/

References

Amadebai, E. (2022). *Decision trees vs. random forests – what's the difference?* https://www.analyticsfordecisions.com/decision-trees-vs-random-forests/

America's health rankings. (2020). https://www.americashealthrankings.org/explore/annual/measure/Suicide/population/suicide%5C_15-24/state/NM

Brownlee, J. (2019). *A gentle introduction to the bag-of-words model.* https://machinelearningmastery.com/gentle-introduction-bag-words-model/

D'Hotman, D., Loh, E., & Savulescu, J. (2020). Ai enabled suicide prediction tools–ethical considerations for medical leaders. *BMJ Leader*, *5*(2).

Kessler, R. C., van Loo, H. M., Wardenaar, K. J., Bossarte, R. M., Brenner, L. A., Cai, T., Ebert, D. D., Hwang, I., Li, J., de Jonge, P., Et al. (2016). Testing a machine-learning algorithm to predict the persistence and severity of major depressive disorder from baseline self-reports. *Molecular psychiatry*, *21*(10), 1366–1371.

Suicide rates by state 2023. (2023). https://worldpopulationreview.com/state-rankings/suicide-rates-by-state

Suicide statistics and facts. save. (2021). https://save.org/about-suicide/suicide-statistics/

Turcan, E., & McKeown, K. (2019). Dreaddit: A reddit dataset for stress analysis in social media. *arXiv preprint arXiv:1911.00133.*