Application of Machine Learning Techniques on Therapist Notes to Assess Suicidal Ideation

Spencer Ebert

March 26, 2020

Abstract

In this project, we seek to understand some of the risk factors for suicidal ideation (SI) by examining notes from psychotherapy sessions. SI is thinking about, considering, or planning suicide without suicidal behavior. While having SI is not always a strong predictor of death by suicide, it is still important to understand because it negatively affects the quality of life for students. We used natural language processing and machine learning to determine how well psychotherapy notes predict self-reported SI. We looked at common co-occurring words for therapy sessions in which the students report high SI and found increases in the word depression result in a higher probability of high SI. Conversely, we found that an increase of the word anxiety resulted in a lower probability of high SI.

1 Introduction

In the past few decades, suicide has grown significantly and affected many people's lives. The issue is especially prominent among college-aged students. Suicide is the second leading cause of death for college students and approximately 12% of college students report having suicidal ideation during their time in school (Centers for Disease Control and Prevention, 2019). Because of the prominence of suicide in this demographic, a majority of students know someone that has attempted or died by suicide.

The purpose of this study is to examine the risk factors related to suicidal ideation for college students who attend psychotherapy. Suicidal ideation (SI) is thinking about, considering, or planning suicide without suicidal behavior. SI may lead to suicidal behavior, but the connection between the two is not as clear as expected. The presence of SI is not necessarily a good indicator of future suicidal behavior according to multiple studies (Rogers, Ringer, and Joiner 2018). However, the importance of studying SI remains strong because alleviating SI will increase the quality of life for those experiencing it whether or not such thinking leads to suicidal behavior.

Many studies have looked into some of the factors that influence SI including drug abuse, anxiety, academic pressure, and isolation (Arria et al. 2009). However, the risk and protective factors that contribute to or alleviate SI are complex and often difficult to study. One source of data available to study SI comes from responses to mental health self-outcome questionnaires. Self-outcome questionnaires are studied instruments that measure mental health. They are completed by the person receiving treatment before each therapy session. One example of a self-outcome questionnaire is the Outcome Questionnaire 45 discussed in the next section.

Recently, natural language processing (NLP) has been to be used to gain further understanding from unstructured sources of text data. For example, recent studies have used NLP on electronic health records (EHRs) to auto-identify problem usage of prescription opioids (Carrell et al. 2015) and suicide attempts (Fernandes et. al 2018). Using similar NLP methods, we have a rich data source found in case notes by therapists from psychotherapy sessions. NLP provides a framework to learn from unstructured data to identify overall themes and connections. Delving into case notes provides a qualitative response to a student's struggles with suicidal ideation, potentially adding insight to the quantitative data from self-report outcome questionnaires. Factors like family relationships, friends, personal struggles and therapist opinions can be found by utilizing the case notes.

We have two main research goals for this analysis (1) How effective are the case notes in predicting self-reported SI? (2) What terms in the case notes have the highest contribution to predicting self-reported SI?

2 Data

2.1 OQ 45

The Outcome Questionnaire-45 (OQ-45) is a self reported rating scale, consisting of 45 questions, used to study the effectiveness of psychotherapy (Lambert et. al 1996). It is one of the most widely used self-report outcome measures in use. This project uses OQ-45 data gathered from the BYU Center of Counseling and Psychological Services (CAPS). Every time a student comes into CAPS for a counseling and/or psychotherapy appointment he/she fills out the OQ-45. The response to each question is on a scale from 0 to 4 (Never, Rarely, Sometimes, Frequently, Almost Always). Some example questions include: "I tire quickly"; "I feel stressed at work/school"; "I am a happy person". For this analysis we focus on question 8 (OQ 8), "I have thoughts of ending my life", the only question on the OQ-45 that specifically asks about SI. We defined individuals who responded "Never" or "Rarely" to OQ 8 as having low SI, and individuals who responded "Sometimes", "Frequently", or "Almost Always" to OQ 8 as having high SI. This labeling of low or high SI is the response variable in our machine learning model.

2.2 Case Notes

The next set of data used for the analysis also comes from CAPS and it contains the case notes for each psychotherapy appointment. For every appointment a student has, the therapist writes impressions of the student's mental health, and a plan moving forward. The therapist also writes quotes from the student explaining current events the student is going through (e.g. struggles in marriage, church, relationships).

The case notes provide qualitative responses regarding the student's life. Instead of relying on responses from a questionnaire, we can look at more in-depth factors that play into SI. For example, we can look at if the person is struggling with anxiety or has any current events in their life that are contributing to SI.

We merge the two datasets to provide both SI level computed from the response to OQ 8 and the factors to predict SI from the case notes. After filtering out non-individual and unattended appointments, the merged data contains 148,297 observations for 19,664 individuals spanning from 2007 to 2020. Over 50% of the students had 4 or fewer appointments. For this analysis, we assume that appointments are independent.

3 Methods

3.1 Natural Language Processing

One of the obstacles from the case notes is the unstructured nature of text. The computer is not able to read the raw case notes and draw the same conclusions that we gather from the notes. To overcome this, we use natural language processing (NLP) to structure the data into meaningful pieces that can be used in the analysis.

NLP provides a method to analyze text. The main idea for NLP is to turn text into meaningful numerical results that can be used in a model. Some examples of numerical results from text can include word counts, bigram counts, and term frequency. Bigrams are tokens of two words next to each other. There is also software available for specific kinds of text that can look at patterns found in that particular genre. For the purposes of this analysis, we use term frequency for single words and bigrams.

The first step in NLP is to clean the text. We cleaned the case notes by eliminating numbers, punctuation, and extra spaces. Furthermore, we eliminated stop words contained in the qdap Top200 dictionary. Stop words are words that are commonly used in everyday language. Some examples include “a”, “the”, “and”, “but”, and “to”. Since these words are common we do not want to include them in the analysis. By getting rid of stop words and nonessential pieces of the document, we are able to focus more on impactful words that provide insight into factors that predict SI.

The next step of NLP is tokenization of the text. Tokenizing the text means breaking the text into specific words or phrases for each individual observation. We tokenized the text looking at single words and bigrams. The purpose of bigrams in this analysis is to capture more meaning from the combination of two words. For example, if the case notes said “not depressed,” the bigram would be able to capture the context rather than just including “depressed.”

Table 1: Example of bag of words matrix. The first observation comes from the example excerpt.

Case Notes	marriage	intent	suicidal	depression	client	family	instance experiencing
1st Observation	0	1	1	2	0	0	1
2nd Observation	1	0	2	0	0	1	0
3rd Observation	0	0	0	3	1	0	0

Following is an excerpt from one of the case notes. We use it as an example of how the text would be cleaned and tokenized. We don't include the date for privacy reasons.

> He reported one instance in [Date] (when he was experiencing depression) when he had a suicidal thought. He reported no plan or intent at that time and sought professional support at that time to help him with his depression.

After cleaning the text, we are left with the following words.

> reported instance experiencing depression suicidal thought reported no plan intent sought professional support depression

From this excerpt, we can tokenize the text and notice that there are two cases of "depression" and one case of "suicidal". Some of the bigrams would include "reported instance", "instance experiencing", and "experiencing depression".

The tokenization process is done for all the case notes. From the cleaning and tokenization, we are able to create a bag of words which is a matrix with the observations on the rows and each token found in the case notes forming a column. The columns are the count of each token in that particular document. From the previous example, we would have an observation with the number of columns being the total number of tokens from all the text. The column for "depression" would have 2, "intent" would have 1, "instance experiencing" would have 1, and so forth. The rest of the entries would be 0. Table 1 shows an example of what a piece of the matrix looks like after tokenization.

Instead of plain word counts, it is common in NLP to use Term frequency-Inverse document frequency (TF-IDF). TF-IDF is a measure of how important a word is to a document in a collection of documents. The calculation is composed of two parts (1) the frequency of a token within a document and (2) the frequency of a token across all documents.

The term frequency (TF) takes into account the size of a document. A 20-word document that contains the word "depression" 5 times is different from a 1000-word document containing "depression"

5 times. Rather than just reporting 5 for both cases, we divide the word count by the total number of words in the document.

We also want to up-weight words that are infrequent but meaningful as compared to words that are more common. This is where inverse document frequency (IDF) comes in. If a particular word is only used in some of the documents, then it may have higher importance than a word used more frequently so we up-weight that particular term in the matrix.

The formula for TF-IDF is given in equation 1.

$$\frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \times \log \left[\frac{N}{|\{d \in D : t \in d\}|} \right] \tag{1}$$

Where $f_{t,d}$ is the word count of term t in document d, $\sum_{t' \in d} f_{t',d}$ is the total number of words in the document, N is the total number of documents, and $|\{d \in D : t \in d\}|$ is the number of documents the term t appears in.

The last step of the process is to prune the tokens in our matrix. There are specific words often used in the case notes that are not considered stop words that we want to prune. Some of these words include “reported” or “client”. Likewise, we want to prune words that are used infrequently like specific names or places. Because these words are used in almost every document or only rarely, we prune them so we can focus on impactful words to predict SI. The specific process we used for pruning the vocabulary is further discussed in the tuning parameters section.

After performing NLP, we ended up with a 122,148 by 1,550 matrix with the 1,550 pruned terms for each document. From this matrix, we predict suicidal ideation which is defined from the student’s response on the OQ 8. We use eXtreme Gradient Boosting to examine the relationship between the bag of words and the OQ 8 response and is discussed in the next section.

3.2 Decision Trees

The relationships between the factors in the case notes and SI are complicated and may not be straightforward so machine learning techniques are used to capture these complex relationships. Algorithms like neural networks, support vector machines, random forests, and boosting are used to predict values from complicated datasets. These algorithms are effective in finding complicated relationships between predictors and responses. Of the machine learning techniques, Extreme Gradient Boosting (XGBoost) has seen a surge in popularity. XGBoost effectively utilizes gradient boosting to combine weak learners for an overall prediction. XGBoost is lauded as one of the fastest techniques compared to the others and has been used to win multiple Kaggle competitions(Chen and Guestrin, 2016).

To understand the XGBoost algorithm, one must first have a basic understanding of decision trees. The main idea behind a decision tree is to split the predictor space $\boldsymbol{X}$ into K partitions and assign a value to each observation in different partitions that predicts the response Y. The process of splitting the space into K partitions uses recursive binary splitting. Basically, all observations are grouped together at the top and will get split into two spaces based on some criteria for the first step. This process of splitting into two groups (binary) continues for the remaining steps (recursive).

To predict a response y_i, each partition of the space is given a constant c_k. Each observation x_i that lands in that particular partition is assigned c_k as the constant, which is the prediction for the response. This is represented in equation 2 where $f(x_i)$ is the predicted value given x_i and R_k is the k^{th} partition.

$$f(x_i) = \sum_{k=1}^{K} c_k I(x_i \in R_k) \tag{2}$$

Since it is computationally infeasible to consider every possible partition of the predictor space, we utilize recursive binary splitting to find an optimum partition. We start with every observation in the same space and then split into two spaces R_1 and R_2 based on criteria for the j^{th} predictor of $\boldsymbol{X}$ based on its value s.

$$R_1(j, s) = \{\boldsymbol{X} | \boldsymbol{X}_j \leq s\} \quad \text{and} \quad R_2(j, s) = \{\boldsymbol{X} | \boldsymbol{X}_j < s\} \tag{3}$$

To decide on the best splitting criteria, the loss function is minimized for all possible combinations of j and s. For the XGBoost algorithm, we use cross-entropy for the loss function which is defined in equation 4.

This process is then repeated for resulting partitions R_1 and R_2 and continued for the desired number of partitions. One of the main challenges of decision trees is the tendency to overfit the data. Methods like pruning and penalties for complexity to the model are used to help mitigate the risk. For XGBoost, we prevent overfitting by specifying the number of splits for each tree and that parameter is tuned in section 3.4.

3.3 Gradient Boosting

The main idea behind gradient boosting for classification is to combine weak learners such as a small decision tree to create a strong learner. Weak learners are algorithms that do slightly better at predicting values than by pure chance. Based on previous weak learners, we build another weak learner and combine the two. The process of building on the previous process is referred to as boosting. The overall goal is to train an algorithm that can take a new observation and predict the class based on the gradient boosted trees.

Other methods also use the combination of multiple trees to come up with a prediction. Adaptive boosting (AdaBoost) is an example where each successive tree is built from the resulting tree by increasing weights for observations that were missclassified in the previous tree. Gradient boosting instead builds trees based on the residuals of the previous tree. The residuals in the case for classification is the observed value minus the predicted probability. By building a new tree based on the previous trees residuals, the algorithm focuses on where the previous step didn't perform well. By recursively building trees, the residuals will be reduced and the model will fit the data better.

Gradient boosting starts with an initial prediction value $F_0(x)$ that is calculated by minimizing the loss function for the current model. In the case for classification, the loss function is defined in equation 4.

$$L(y, p(x)) = -(y\log(p(x)) - (1-y)\log(1-p(x))) \tag{4}$$

This loss function is the deviance where $p(x)$ is the probability of the observation having SI and y is 1 if the observation had suicidal ideation and 0 otherwise.

From this initial step $F_0(x)$, we want to move in a direction that minimizes the loss and improves where the initial guess didn't perform well. To do this, pseudo residuals are calculated. The residuals are equal to the negative partial derivative of the loss function with respect to $F_0(x)$ and probability $p(x)$ in the loss function equal to the previous prediction $F_0(x)$ (See step 2.a in the following algorithm). We fit a decision tree to the residuals as discussed earlier to partition the predictor space into R_k groups. From these groups we assign a value $\gamma_{j,m}$ to each resulting partition in the residual tree, which is calculated in equation 5.

$$\hat{\gamma}_{j,m} = \arg\min_{\gamma_{j,m}} \sum_{x_i \in R_{j,m}} L(y_i, F_{m-1}(x_i) + \gamma_{j,m}) \tag{5}$$

$F_{m-1}(x_i)$ is the predicted algorithm for the previous step given a value x_i.

With that new value calculated we update the prediction function and update again. A technical description of the algorithm is given as follows.

1. Initialize model with a constant value $F_0(x) = \arg\min_\gamma L(y_i, \gamma)$

2. for m in $1, ..., M$

   (a) Compute pseudo-residuals $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F=F_{m-1}}$ for $i = 1, ..., N$

   (b) Fit a decision tree to the targets r_{im} giving terminal regions $R_{jm}, j = 1, 2, ..., J_m$

   (c) For $j = 1, ..., J_m$ compute $\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma)$

   (d) Update $F_{m,g}(x) = F_{m-1,g}(x) + \nu \sum_{j=1}^{Jm} \gamma_{jm} I(x \in R_{jm})$

3. Output $\hat{F}(x) = F_M(x)$

In the algorithm ν is a tuning parameter often referred to as the learning rate. We do not want to overfit the data so instead of taking big steps in the right direction we take a lot of small steps of size ν in the correct direction. Small values of ν around 0.1 are common, but this is one of the tuning parameters we tune as described in the next section.

By taking the gradient with respect to the previous function at each step (Step 2.a), we move in the direction that minimizes the current loss function. XGBoost follows the gradient boosting framework, and utilizes regularization, parallel computing, and the second derivative of the loss function to speed up the process, and help reduce the chance of overfitting.

3.4 Tuning Parameters

One necessity with some machine learning techniques including XGBoost and NLP is choosing tuning parameters for the algorithms. The choice of tuning parameters can largely influence the overall outcome. There were six parameters we tuned for the analysis. For NLP, we tuned parameters for minimum word count, document proportion minimum, and document proportion maximum. The minimum word count gets rid of all words that appeared less than a certain threshold across all documents. Document proportion minimum gets rid of words that appeared in a small proportion of the documents. Document proportion maximum gets rid of words that appeared in a large proportion of the documents. The three parameters we tuned for XGBoost are the learning rate ν, the maximum depth of the decision trees, and the number of rounds M. The different parameter values we checked are found in table 2. We looked at all possible combinations of the chosen parameters which resulted in 729 possible parameter combinations.

Table 2: Tuning Parameter Values

Tuning Parameters	Values to look at
minimum word count	**50** 100 300
document proportion min	**0.01** 0.1 0.2
document proportion max	**0.5** 0.7 0.9
learning rate ν	**0.01** 0.1 0.3
maximum depth	2 5 **10**
number of rounds	50 100 **1000**

The machine learning pipeline we followed to estimate predicted SI from case notes and choose tuning parameters is as follows.

1. Split the data into three sets 50% train, 25% validation, and 25% test.

2. Oversample training data to even out suicidal ideation response.

3. Choose tuning parameters as follows.

    (a) For each combination of tuning parameters, prune the vocabulary, and fit the XGBoost model using the oversampled training data.

    (b) Predict SI for the validation set.

    (c) Compute F1 rate on validation set. Equation found in equation 6.

    $$F1 = 2\left(\frac{precision * recall}{precision + recall}\right) \tag{6}$$

    Precision is the proportion of values predicted to be true that are true (positive predictive rate). Recall is the proportion of values that are true that were predicted to be true (sensitivity).

    (d) Choose the tuning parameters that give the highest F1 rate.

4. Refit model with chosen tuning parameters.

5. Calculate final F1 values on the testing dataset.

We choose to split the data into three sets to help prevent overfitting in the model. We want to look at how well our model works for observations that were not used to build the model. The validation data is used to compute out-of-sample F1 rates to determine the best tuning parameters. The overall prediction performance should be calculated with data that were not used to tune the parameters or originally fit the model. That is where the test data comes in. We can now look at prediction results from the test data that were not used in the process of tuning parameters.

One of the data issues we had to address with this dataset is that of class skew. That is, about 83% of the study participants were identified as having low SI and the other 17% as having high SI. Because of the smaller percentage of people labeled with high SI, we randomly sampled high SI observations on the training data with replacement to get the proportion closer to 50%. This helps the model focus equally on high and low SI. We do not perform oversampling on the validation and test datasets because we want to maintain the same proportion when looking at prediction performance.

Because the validation dataset maintains the class skew, we choose tuning parameters based on the F1 rate. If we were to choose tuning parameters based on the correct prediction rate we would be tend to choose models that favor predicting low SI. The F1 rate gives greater importance to balancing false positives and false negatives and is a better measure of model performance than accuracy when there is imbalance.

The resulting tuning parameters for our model are the bold values in table 2.

3.5 SHAP Scores

A common criticism of some machine learning methods is their lack of interpretability. Machine learning algorithms are commonly seen as black box methods. You input an observation and a prediction gets output without knowing exactly how the factors influenced the outcome. Recently there has been a growing focus on interpretable machine learning. Methods like LIME (Ribeiro, Singh, Guestrin 2016) have been studied to interpret results from machine learning models. Since one of our research questions require interpretability, we utilize current methods that look at individual factor contributions to a prediction. In this analysis, we use SHapley Additive exPlanations (SHAP) (Lundberg, Lee 2017). The idea of SHAP comes from the game theory concept of Shapley values. Shapley values measure individual contributions of factors to an outcome. For each observation, there is a SHAP score on each factor in the observation. Basically, every observation has different SHAP scores for the same factors.

One observation may have put more importance to one factor as compared to another observation resulting in a different SHAP score. This results in a 122,148 by 1,550 matrix matching the size of the bag of words matrix used to fit the model.

One of the advantages to SHAP is the ability to look at global effects of a particular factor on the outcome. While the exact SHAP score may not be the same from observation to observation, the general trend can be examined. For example, higher frequencies of the word "suicidal" in the case notes tends to increase the probability the person was labeled with high SI. The SHAP scores show that correlation which can be seen in figure 1.

Another advantage of using SHAP scores is the ability to look at individual observations. If one was curious why a particular observation had a higher probability of having high SI, they can look at the individual SHAP scores for that observation and see what factors had the biggest influence on the prediction.

4 Results

For this analysis, we did two iterations of NLP. On the first iteration, we pruned based on the tuning parameters found earlier. The second pruning used the same parameters from the first but also excluded the words "suicidal", and "suicide". The reason we excluded those words is because we wanted to see if the prediction performance changed dramatically eliminating those words, and we wanted to see if any new words popped up as being important in the model. Results for both methods are given in the following two sections.

4.1 Prediction Performance

From the models built, we are able to predict SI for people based on their case notes. To assess the predictive performance of our model we look at the prediction performance on the test dataset. Tables 3 and 4 give the confusion matrices, F1 rates, and correct prediction rates for both models.

For both cases, the F1 rates from these models are not very high. Using the case notes to predict a student's response to OQ 8 shows some merit from these prediction rates, but not a perfect method of prediction.

Table 3: Including the words suicidal and suicide

	Predicted low SI	Predicted high SI
True low SI	27,460	3,277
True high SI	3,383	2,955

F1	47%
Correct Prediction	82%

Table 4: Excluding the words suicidal and suicide

	Predicted low SI	Predicted high SI
True low SI	27,371	3,366
True high SI	3,586	2,752

F1	44%
Correct Prediction	81%

Looking into the notes that were not predicted correctly helps understand why the model did not capture some of the OQ responses correctly. Some of the notes that were predicted as having high SI when the student reported having low SI are a result of the student talking about someone they know that has high SI. The tokenization for our NLP has a hard time distinguishing if the case notes are talking about the particular person or if they are talking about someone else. There are also case notes where the therapist says that the student denied having high SI but the therapist believes that they are at risk despite the student denying it. This gives some credence to using the case notes to identify students with high SI. Rather than relying purely on the student's response for their SI, we can rely on a qualitative evaluation from an expert.

4.2 Word Contributions

Another purpose of this analysis was to see which words had the highest influence on predicting suicidal ideation. Figures 1 and 2 show SHAP scores for the top 15 most influential terms. Figure 1 looks at the model including all words while figure 2 looks at the model excluding suicide and suicidal.

These graphs look at SHAP values for every observation (i.e. case note) for the top 15 most influential terms. Each point represents an observation from the training data. The horizontal axis gives the SHAP value for each observation. The color corresponds to the actual feature value. For this analysis, a point is yellow if the TF-IDF is close to 0 and is purple for higher TF-IDF values. In other words, purple positive values show that an increase of the particular term results in higher probability of high SI, and purple negative values show that an increase of the particular term results in higher

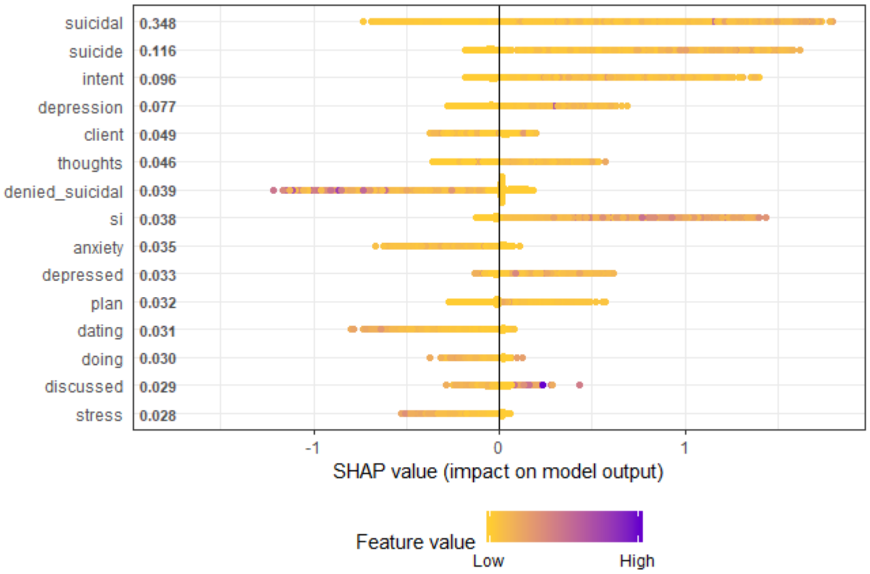

Figure 1: Top 15 SHAP scores including the word suicidal

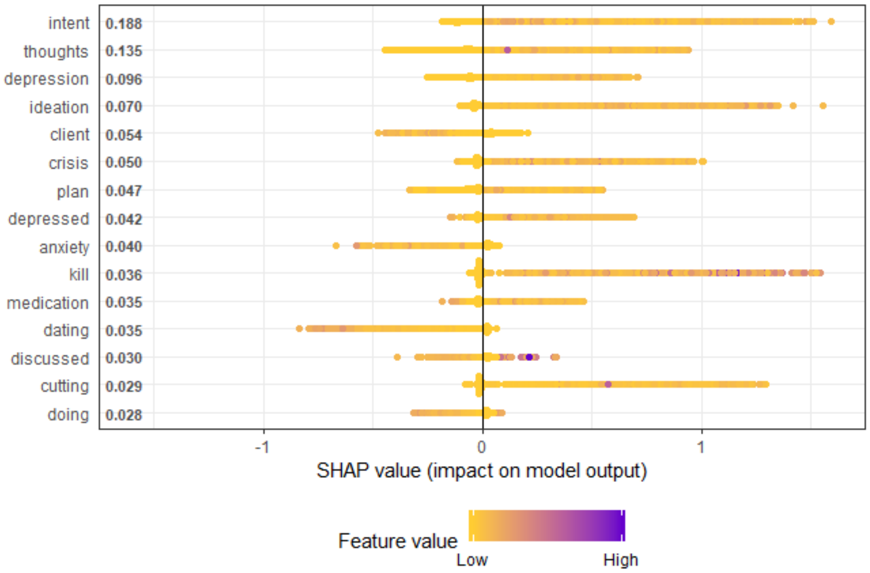

Figure 2: Top 15 SHAP scores without including the word suicidal

probability of lower SI. Conversely, yellow positive values show that a decrease in the particular term results in higher probability of high SI, and yellow negative values show that a decrease in the term results in lower probability of high SI.

Some of the key findings from the SHAP scores are shown below.

1. The model that includes the terms suicide and suicidal has those two words as the most important for predicting SI. Greater frequencies of those words results in a higher probability of a person having high SI. That result matches intuition from the case notes predicting SI.

2. The most influential factors are largely the same for both models.

3. Depression, intent, plan, and depressed show high positive correlations to predicting high SI. If the case notes include those words more frequently, then the probability of having high SI goes up. Intuitively it makes sense that higher depression tends to lead to higher SI.

4. Interestingly, an increase of the word anxiety in the case notes tends to result in a higher probability of having low SI. This does not necessarily mean that a person with anxiety has a lower SI, but that an increase in the frequency of anxiety in these particular case notes tends to lower the probability of high SI response from the student. This may be due to the fact that depression and anxiety are often seen in conjunction with each other in the case notes, with the word depression having a larger weight than anxiety.

5. An increase of the word dating tends to reduce the probability of a person having high SI. This seems to indicate that higher frequencies of words relating to a relationship result in lower probabilities of having high SI.

It is important to note that these results are not causal relationships and that the response is based on a student's answer to a questionnaire, not the actual diagnosis. However, we can see that there are some important relationships between the case notes written by the therapist and the response the student gives. The results suggest that people with depression, or past history (intent) have higher SI. They also suggest that people with more social connections and a support group tend to have lower SI. It could also mean that people have lower SI because they have someone relying on them. In either case, the results suggest that isolated people that feel alone have higher SI.

5 Conclusion

Overall, we examined the relationships between SI and case notes by utilizing NLP and XGBoost. These machine learning techniques allowed us to look at complex relationships between variables and output predictions based on case notes. This study showed people that have fewer mentions of social connections like dating and higher mentions of depression tend to have higher SI. Even though these results are not causal, they leave room for further analyses into the specific relationships between SI and these factors.

Further, our machine learning algorithms incorrect prediction for high SI were able to capture some discrepancies between a student's response and the therapists evaluation. A student's response on a questionnaire may not be the best indicator of their true SI. Because of this, we plan on labeling some of the case notes based on a risk score of low, medium, or high SI. From this label, we can build a model from the labeled case notes and predict therapists' evaluation of SI of a particular student rather than relying purely on the student's response.

One potential weakness of this analysis is that the chosen tuning parameters are all on the edges. This means that better tuning parameters can potentially be found outside of our chosen grid. Moving forward, we want to examine more ways to tune the parameters instead of relying on a grid search with predefined values. This can include gradient descent for the hyper parameters and would eliminate the need to give a predefined grid of values.

Another potential weakness of this analysis comes from the correlation between observations. We looked at predicting SI for a particular appointment not for an individual. Since some students had multiple appointments there is correlation between the observations. Moving forward, we want to look at identifying SI for individual students and seeing how SI changes over time for students.

Understanding factors that relate to SI is helpful to improving the quality of life for students and decreasing SI. There are many methods and avenues to follow in understanding important relationships. This analysis builds a basis for future work in NLP on therapist notes to examine qualitative relationships for SI.