Sentiment Analysis for Mental Health Monitoring

1st Sri Kalyan Reddy Akiti
dept. Data Science, college of science
Texas A and M university
Corpus Christi, United States
sakiti@islander.tamucc.edu

Abstract

Mental health problems are among the fast-growing challenges facing the world today, and to identify the problem early is crucial for appropriate intervention. This work uses machine-learning methods to classify texts into various mental health categories of interest which include *Anxiety, Bipolar, Depression, Normal, Personality Disorder, Stress, and Suicidal*. The dataset has preprocessed using natural language processing techniques like lemmatization, stop word removal, and TF-IDF vectorization. In addition, Synthetic Minority Over-sampling Technique was implemented to deal with class imbalance. Three machine learning models include Logistic Regression, Random Forest, and Support Vector Machine were developed and evaluated. Random Forest was observed to take the higher note among these models with an accuracy level of about 90% over most categories, while the model reported challenges measuring specific categories such as Suicidal and Depression due to occasional bounce in classification. Machine learning promises to monitor health concerning mental well-being, but finer techniques are needed to capture human emotion puzzles. This work paves a way toward building real-time applications for aiding mental health professionals in timely diagnosis and monitoring.

Introduction

The essential ingredient of well-being is mental health-the catch is that it is difficult to recognize or to deal with the mental health issues at an early stage. Most of the expressions are done in writing-some use journals while others post things on social media or talk to friends. These expressions seem to be good sources of clues about the mental state of a person. This project focuses on-machine learning how texts can be analyzed to detect possible mental illnesses, including *Anxiety, Depression, and Stress,* through suicidal thoughts. The objective is well defined by designing a program that registers the early signs of these conditions with valuable aid for mental health professionals.

It was achieved through advanced text cleaning and preprocessing technique on real-world data followed by training of machine learning models including: Logistic Regression, Random Forest and SVM. There was also the challenge of unequal representation of some mental disorders among the types of conditions where one condition was represented more than its usual frequency in the dataset SMOTE was used towards ensuring so-called fairness in predictions. Our findings indicated that model predictions were viable, particularly from the Random Forest model, with specific reference to the mental state of a person. Furthermore, the project has highlighted how technology could be an essential part of future mental health care, making it easier to identify problems and provide support during the most critical times.

Related Work

[1]In recent years, research in the area of employing machine learning for mental health monitoring has escalated. Several studies have applied the techniques of natural language processing (NLP) to analyze data collected from diverse text sources like social media posts, therapy transcripts, and self-reported mental health questionnaires to identify psychological conditions. Sentiment analysis using emotional signs is a critical application for general topics modeling by discovering hidden themes linked with mental states. Standard machine-learning methods such as logistic regression, support vector machine (SVM), and random forest proved efficient for classifying conditions across text-based categories but suffer challenges such as imbalance datasets, symptom overlap, and subtlety in language expression.

The emergence of deep learning models and their usage has pushed the frontier even farther beyond basic deep constructs. Recurrent neural network (rnn), long short time memory (lstm) networks, and the entire transformer-based model families that include BERT proved much better qualitatively as they captured their context-sensitive meaning through the very design of their structures. This is becoming even more powerful and accurate but typically comes at heavy computational costs, particularly when using large sets of data for training that infeasible for applications where resources are restricted. Nevertheless, with all these advancements, interpretations regarding models compared to their performance are still a question mark worth considering, especially in the area of mental health monitoring, where sensitivity is paramount.

In most recent times, new studies adopted techniques like Synthetic Minority Oversampling Technique SMOTE to overcome such challenges with a performance such that the minority classes-the less common ones of the host mental health disorders-would get an adequate representation in training data; through that basis, advanced NLP preprocessing along with machine learning approaches are aligned with the creation of an efficient and interpretable system. This improves classification accuracy but also connects the benefits of technological advancement into the realization in the very domain of mental health monitoring.[2]

Proposed Method

The structured workflow is to create a machine learning system capable of classifying textual data based on different mental health disorders. First, the processing of the data would be done by transforming the entire text to lowercase and stripping it of all punctuations simply, but not stop words or any other insignificant words. Lemmatization would reduce words to their original form while preserving meaning. After preprocessing, feature extraction would be done via a Term Frequency-Inverse Document Frequency (TF-IDF) vectorization method, which converts text into numerical representation capturing unigrams and bigrams as much as possible up to a maximum features size of 5000, that together form a maximum of 5000 features. The class imbalance correction is done using the Synthetic Minority Over-sampling Technique (SMOTE) for creating synthetic samples for minority classes to fill the training dataset. Three machine learning models Logistic Regression, Random Forest, and Support Vector Machine (SVM) would last be trained. They would be compared based on some metrics such as accuracy, precision, recall, F1 score, and confusion matrix as applied to the processed data to measure their effectiveness in differentiating between the mental health conditions. The models would finally be subjected to some predictions about those categories using test statement examples while results would be compared against expected categories to show applicability as well as practical interpretability of the system monitoring mental health in real life.

Dataset

[3]The sentences within the data set included assertions and labels referring to specific fields of mental health: anxiety, bipolar, depression, normal, personality disorder, stress, and suicidal ideation. In the beginning, the data was in CSV format, and the data underwent pre-processing to remove missing and incomplete records to ensure data quality.

The feature is *statement*, the text defining the mental states, while the target variable is the *status* which is the mental health condition. The dataset is imbalanced, with some categories underrepresented, thus affecting training of the machine learning models. SMOTE was employed to address this problem by synthesizing minority class samples into the dataset.

The final dataset has gone through the processes to be able to come up with a clean structure that is amenable to machine learning tasks. It was then split into train-test sets to evaluate the performance of the proposed models. The cleaned and balanced dataset combined with feature extraction techniques made it possible to classify mental health conditions efficiently.

Methodology

The project is doing extensive processes to aid in the classification of mental health conditions from text data by machine learning and natural language processing (NLP). The first step of the entire process will be cleaning the raw text data before subjecting it to analysis. Most raw texts are unstandardized because they do not necessarily adopt lower- and upper-case letters in consistent reading. This is important so that "Happy" and "happy" will be treated by the machine as the same word. This is also followed by the elimination of punctuation marks and special symbols that do not contribute to the classification process. Words such as "and", "the", and "is" refer to stopwords. They are also cut off from qualifying purposes. As a matter of fact, lemmatization-which is the process of simplifying words to their base forms, such as " running" into "run," "better" into "good"-is applied as well. These steps give assurance of purity, consistency, and readiness of the text for subsequent analysis.

Then, the data is prepared and converted into numbers understandable for the model learning machine. Term Frequency-Inverse Document Frequency (TF-IDF) is the required vectorization by which the model can apply for the work. Thus, its effectiveness depends on how well the words or phrases are really important for each text piece determined by how often they occur inside a specific statement compared to the whole set of data. The model focuses more on different or important words for each given taciturn state because it considers the frequency occurrence of these words in the entire corpus. The vectorizer utilizes unigrams and bigrams from an individual word capturing different short phrases to include in the analysis. This way, it can ensure that different words and short phrases are captured by analysis. This is the time of feature extraction that will provide the structured dataset, wherein each statement is expressed as a series of numbers ready for input into machine learning models.

The skewedness of the dataset emerged as one of the hurdles experienced in this process. The mental health manifestations of Anxiety or Depression are substantially more populated, while the data on Suicide and Bipolar disorder are less populated. The outcome of this skewness is that the biased models either favor or neglect the more frequent ones. The resolution for this problem resides in a technique termed Synthetic Minority Oversampling Technique or SMOTE. SMOTE works by the creation of new synthetic instances for the underrepresented classes by interpolation between the existing data points. It ensures balance in the data set, making it possible for each mental condition to be equally captured. The accuracy and fairness of the model must be upheld by balanced data with respect to all categories.

Make training of machine learning models on the balanced dataset. Three different models had to be chosen for this purpose: Logistic Regression, Random Forest, and Support Vector Machine (SVM). These models were chosen for being good performers for most classification applications and have their pros and cons. Logistic Regression also makes for a simple and interpretable baseline reliability model. Random Forest is composed of some decision trees working together for very complicated patterns of data. SVM is yet another type of classifier that does well against high dimensional spaces and is especially suitable for text data. The data were split into training and testing subsets, in which 80 percent of the data were dedicated to training the models and then 20 percent for the testing of their performance. This split ensures that the models are evaluated on unseen data and thus diminishing the risk of overfitting.

The effectiveness of the models has been appraised using a number of performance metrics; those include accuracy, which measures the overall correctness of the model; precision, which gives the ratio of correct identification of exactly which of the specific mental health condition with normal misclassification; recall, which measures the model's ability to find all relevant cases of each condition; and the F1-score, which is a trade-off between the precision and recall. Moreover, the confusion matrices are generated for each model, providing a well-structured break-up of amounts in true positives, false positives and false negatives for each mental health condition. Such evaluations would help decide for each model amongst the best avenues taken while analyzing its quality.

The final testing for the trained models was done using example statements that highlighted their practical usefulness. Such statements characterized various mental health conditions, such as "I feel like something bad will happen soon:" that should be classified as "Anxiety," or "I feel empty and struggle to find joy:" that corresponds with "Depression." They were then preprocessed and vectorized as the data training. The model predictions were subsequently compared with the expected categories to test their validity. Out of the three models Random forest showed the best performance concerning test data among all three. So Random forest is a model that can better classify very complex and subtle form of text data. The prediction explained the strength of the system clearly along with the areas where there is potential improvement.

Such methodology provides, in an organized and complete manner, the utilization of machine learning on mental health classification. The final system along with its accuracy and interpretation results is possible by paying due consideration while cleaning and preparing data, addressing class imbalance, and evaluation of several models. It makes real-world example statements possible, which have great potential in being utilized by mental health professionals to facilitate the detection and monitoring many mental health conditions. This system is efficacious for early diagnosis and intervention.

Here is all the methodology, organized and complete, for the use of machine learning in mental health classification. Cleaning and preparing data, attending to class imbalance, and testing several models will allow for the creation of a final system that is both accurate and interpretable. Such real-world example statements could be a powerful addition to the arsenal of mental health professionals who use them to uncover and track many mental health conditions; thus, this system is effective for early diagnosis and intervention.

Conclusion

Indeed, this effort evokes that machine learning and NLP will appropriately fit into the classification of text regarding mental health disorders. Challenges that comes with real-world mental health data are effectively handled by a systematic approach involving robust preprocessing, strong feature extraction, and balancing techniques. As an example, Term Frequency-Inverse Document Frequency (TF-IDF) vectorization made feature extraction from text interesting, and Synthetic Minority Oversampling Technique (SMOTE) ensured equal representation of all mental health categories for model training. Such basic steps become crucial as very little was needed for machine learning models to make fair and accurate projections.

Comperision of Logistic Regression, Random Forest and Support Vector Machine (SVM) gave a good clear idea about the merits and demerits of the models. Among all, Random Forest was found to be the most accurate and reliable classifier, scoring 90 percent accuracy, while Logistic Regression and SVM also showed quite good performance with above 80 percent. However, it can still be found some challenges like classifying subtle or crossing categories like Stress and Anxiety, or rare conditions such as Suicidal, etc. The limitations point towards the need for adopting advanced methods like deep learning models or some domain-specific embeddings for better capture of complex language patterns.

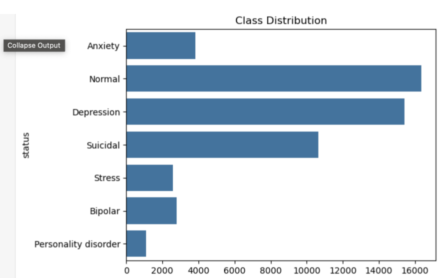
In simpler terms, the findings of this project show that it is indeed possible to monitor mental health using technology. The system cannot be expected to replace human mental health professionals; however, it can be useful as an initial detection and support tool. The systems can then be integrated into the health workflow to scale mental health services while reaching those who might otherwise go undiagnosed. Future work includes refinement of current models, building them with larger datasets, and furthering their interpretability toward the development of applications that are more effective and ethical with AI in practice within the mental health field.

In less complex terms, results from this project would indicate that it is indeed possible to monitor mental health via technology; that is not to say that this kind of system will replace human professionals in mental health. It can, however, work as a useful first step in the detection and support process. Embedded into the healthcare workflow, these systems become scalable for mental health services to reach those who would otherwise not be diagnosed. Future work will include further refinement of current models, establishing them with larger datasets, and increasing their interpretability toward creating applications in mental health practice more effective and ethical in the use of AI.

Appendix

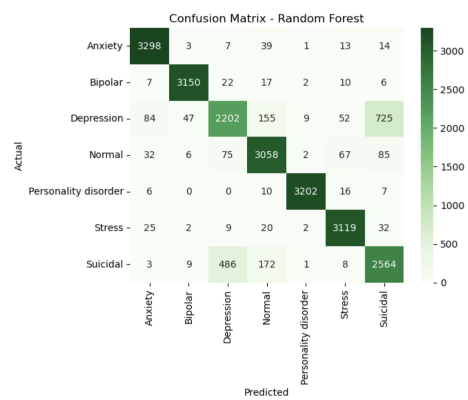
```python
# Import necessary libraries
import pandas as pd
import numpy as np
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from imblearn.over_sampling import SMOTE
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import nltk

# Download necessary NLTK data
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

# Load the dataset
df = pd.read_csv('Data.csv', index_col=0)

# Drop missing values
df.dropna(subset=['statement'], inplace=True)

# Define text preprocessing function
def preprocess_text(text):
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words('english'))
    text = text.lower()  # Convert to lowercase
    text = text.translate(str.maketrans('', '', string.punctuation))  # Remove punctuation
    words = [lemmatizer.lemmatize(word) for word in text.split() if word not in stop_words]  # Lemmatize and remove stopwords
    return ' '.join(words)

# Preprocess statements
df['cleaned_statement'] = df['statement'].apply(preprocess_text)

# Visualize class distribution
sns.countplot(y=df['status'])
plt.title('Class Distribution')
plt.show()
```


```python
# Import necessary libraries
import pandas as pd
import numpy as np
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from imblearn.over_sampling import SMOTE
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import nltk

# Download necessary NLTK data
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

# Load the dataset
df = pd.read_csv('Data.csv', index_col=0)

# Drop missing values
df.dropna(subset=['statement'], inplace=True)

# Define text preprocessing function
def preprocess_text(text):
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words('english'))
    text = text.lower()  # Convert to lowercase
    text = text.translate(str.maketrans('', '', string.punctuation))  # Remove punctuation
    words = [lemmatizer.lemmatize(word) for word in text.split() if word not in stop_words]  # Lemmatize and remove stopwords
    return ' '.join(words)

# Preprocess statements
df['cleaned_statement'] = df['statement'].apply(preprocess_text)

# Visualize class distribution
sns.countplot(y=df['status'])
plt.title('Class Distribution')
plt.show()
```


```
Training Logistic Regression...
Training Random Forest...

Logistic Regression Classification Report:
                      precision    recall  f1-score   support

             Anxiety       0.90      0.85      0.87      3375
             Bipolar       0.95      0.95      0.95      3214
          Depression       0.74      0.62      0.68      3274
              Normal       0.65      0.81      0.72      3325
Personality disorder       0.91      0.89      0.90      3241
              Stress       0.82      0.81      0.82      3209
            Suicidal       0.70      0.71      0.70      3243

            accuracy                           0.81     22881
           macro avg       0.81      0.81      0.81     22881
        weighted avg       0.81      0.81      0.81     22881
```

```
Random Forest Classification Report:
                      precision    recall  f1-score   support

             Anxiety       0.95      0.98      0.97      3375
             Bipolar       0.98      0.98      0.98      3214
          Depression       0.79      0.67      0.72      3274
              Normal       0.88      0.92      0.90      3325
Personality disorder       0.99      0.99      0.99      3241
              Stress       0.95      0.97      0.96      3209
            Suicidal       0.75      0.79      0.77      3243

            accuracy                           0.90     22881
           macro avg       0.90      0.90      0.90     22881
        weighted avg       0.90      0.90      0.90     22881
```




```
Logistic Regression Accuracy: 0.8053
Random Forest Accuracy: 0.9000

Predictions with Logistic Regression:
Original Statement: My heart races, and I feel like something bad will happen soon.
Actual Label: Anxiety
Predicted State: Anxiety

Original Statement: I have two thoughts on one work.
Actual Label: Bipolar
Predicted State: Normal
```

```
Predictions with Random Forest:
Original Statement: My heart races, and I feel like something bad will happen soon.
Actual Label: Anxiety
Predicted State: Normal

Original Statement: I have two thoughts on one work.
Actual Label: Bipolar
Predicted State: Normal

Original Statement: I feel empty and struggle to find joy in anything these days.
Actual Label: Depression
Predicted State: Depression

Original Statement: I'm feeling good.
Actual Label: Normal
Predicted State: Normal

Original Statement: I often feel misunderstood and find it hard to maintain close relationships.
Actual Label: Personality disorder
Predicted State: Normal

Original Statement: My workload is too heavy, and I can't seem to keep up with everything.
Actual Label: Stress
Predicted State: Normal

Original Statement: I am about to die.
Actual Label: Suicidal
Predicted State: Normal
```

```
[17]: # Import necessary libraries
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report
from imblearn.over_sampling import SMOTE
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import nltk
import string

# Download NLTK data
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

# Preprocessing function
def preprocess_text(text):
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words('english'))
    text = text.lower()
    text = ''.join([char for char in text if char not in string.punctuation])
    words = [lemmatizer.lemmatize(word) for word in text.split() if word not in stop_words]
    return ' '.join(words)

# Load and reduce the dataset size (assuming df is already loaded)
df_reduced = df.sample(n=10000, random_state=42)

# Preprocess the reduced dataset
df_reduced['cleaned_statement'] = df_reduced['statement'].apply(preprocess_text)

# TF-IDF Vectorization
vectorizer = TfidfVectorizer(max_features=1000, ngram_range=(1, 2))
X_reduced = vectorizer.fit_transform(df_reduced['cleaned_statement'])
y_reduced = df_reduced['status']

# Handle class imbalance using SMOTE
smote = SMOTE(random_state=0)
X_resampled_reduced, y_resampled_reduced = smote.fit_resample(X_reduced, y_reduced)

# Train-test split
X_train_reduced, X_test_reduced, y_train_reduced, y_test_reduced = train_test_split(
```

```
# Train the SVM model
svm_model_reduced = SVC(kernel='linear', probability=True, random_state=42)
print("Training SVM on reduced dataset...")
svm_model_reduced.fit(X_train_reduced, y_train_reduced)

# Make predictions
y_pred_reduced = svm_model_reduced.predict(X_test_reduced)

# Display classification report
classification_report_reduced = classification_report(y_test_reduced, y_pred_reduced)
print("Classification Report on Reduced Dataset:")
print(classification_report_reduced)

# Define example statements
example_statements = {
    "Anxiety": "My heart races, and I feel like something bad will happen soon.",
    "Bipolar": "I have two thoughts on one work.",
    "Depression": "I feel empty and struggle to find joy in anything these days.",
    "Normal": "I'm feeling good.",
    "Personality disorder": "I often feel misunderstood and find it hard to maintain close relationships.",
    "Stress": "My workload is too heavy, and I can't seem to keep up with everything.",
    "Suicidal": "I am about to die."
}

# Predict using SVM
print("\nPredictions with SVM:")
for label, sentence in example_statements.items():
    processed_text = preprocess_text(sentence)  # Preprocess the text
    vectorized_text = vectorizer.transform([processed_text])  # Vectorize the preprocessed text
    predicted_status = svm_model_reduced.predict(vectorized_text)[0]  # Predict using SVM
    print(f"Original Statement: {sentence}")
    print(f"Actual Label: {label}")
    print(f"Predicted State: {predicted_status}\n")

[nltk_data] Downloading package punkt to
[nltk_data]     /Users/srikalyanreddyakiti/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/srikalyanreddyakiti/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/srikalyanreddyakiti/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
Training SVM on reduced dataset...
```

```
[nltk_data]     /Users/srikalyanreddyakiti/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
Training SVM on reduced dataset...
Classification Report on Reduced Dataset:
                      precision    recall  f1-score   support

             Anxiety       0.84      0.78      0.81       605
             Bipolar       0.95      0.98      0.97       639
          Depression       0.71      0.57      0.63       634
              Normal       0.64      0.83      0.72       662
Personality disorder       0.96      0.92      0.94       641
              Stress       0.91      0.94      0.92       599
            Suicidal       0.67      0.64      0.66       612

            accuracy                           0.81      4392
           macro avg       0.81      0.81      0.81      4392
        weighted avg       0.81      0.81      0.81      4392


Predictions with SVM:
Original Statement: My heart races, and I feel like something bad will happen soon.
Actual Label: Anxiety
Predicted State: Anxiety

Original Statement: I have two thoughts on one work.
Actual Label: Bipolar
Predicted State: Normal

Original Statement: I feel empty and struggle to find joy in anything these days.
Actual Label: Depression
Predicted State: Depression

Original Statement: I'm feeling good.
Actual Label: Normal
Predicted State: Anxiety

Original Statement: I often feel misunderstood and find it hard to maintain close relationships.
Actual Label: Personality disorder
Predicted State: Normal

Original Statement: My workload is too heavy, and I can't seem to keep up with everything.
Actual Label: Stress
Predicted State: Anxiety

Original Statement: I am about to die.
Actual Label: Suicidal
Predicted State: Suicidal
```

REFERENCES

[1] Saif M Mohammad and Peter D Turney. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. 2016.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.