Table of Contents

Abstract

The prevalence of mental health challenges in today's digital age necessitates innovative methods for timely detection and support. This study leverages the power of natural language processing (NLP) and social media data, particularly Twitter, to identify early signs of mental health issues. By recognizing patterns indicative of mental health conditions like depression, anxiety, or stress, the project identifies at-risk individuals early in their struggles. The analysis of various models, including Naive Bayes, Support Vector Machine (SVM), LSTM, and BERT, underscores BERT's superiority due to its consistent performance across precision, recall, F1-score, and accuracy metrics. BERT's contextual language understanding and adaptability across different data sources, exemplified by its success with Reddit data, position it as a robust solution for this task. MarkovML has proven invaluable by seamlessly integrating bookkeeping, facilitating nuanced comparisons, and enhancing exploratory data analysis, significantly streamlining project workflows and bolstering informed decision-making processes.

Keywords: mental health, tweets, nlp, bert

Introduction

The modern era is characterized by the increasing interconnectedness of our lives with the digital world. This has led to an alarming rise in mental health challenges, such as depression, anxiety, and stress. These invisible struggles underscore the urgent need for innovative approaches to provide timely support and intervention. Approximately 450 million people worldwide are mentally ill, accounting for 13% of the global disease burden. The World Health Organization (WHO) estimates that one in four individuals experiences mental disorders at some point in their lives. Mental health problems can have a severe impact on individuals, their families, and society as a whole. Traditional methods of mental health detection typically involve face-to-face interviews, self-reporting, or questionnaire distribution. However, these methods are often labor-intensive and time-consuming. In the past decade, social media has transformed how people interact with each other. Apart from sharing factual information and news, people actively share their day-to-day activities, experiences, feelings, opinions, hopes, desires, and emotions online. This has created a new opportunity for mental health detection and intervention. A recent research study presented a novel approach to mental health problem detection in online social networks. The study used natural language processing (NLP) and text mining techniques to analyze text posts from social media users. The study was able to identify users who were at risk for mental health problems with a high degree of accuracy. This study

suggests that social media can be used to detect and intervene in mental health problems at an early stage. This is important because early intervention can improve outcomes for people with mental health problems. However, it is important to note that social media is not a substitute for professional mental health care. If you are struggling with mental health problems, it is important to seek help from a qualified professional.
This project builds on this research by harnessing the power of natural language processing (NLP) and social media data, specifically Twitter, to pave the way for early identification and assistance for those silently battling mental health issues. By employing advanced NLP techniques, we aim to decode the intricate language patterns, sentiments, and contextual cues embedded within social media posts. In doing so, we can discern the early indicators of mental health challenges. This, in turn, paves the way for not only identifying those in need but also connecting them with the appropriate resources and assistance. This project envisions a future where individuals who might otherwise suffer in silence can find solace and guidance with a simple social media post. By identifying these individuals early in their struggles, we can break the barriers of stigma and isolation, providing timely interventions that can make all the difference. The seamless integration of the Markov platform has profoundly enriched our study's operations. Through its toolkit, we've conducted meticulous exploratory data analysis, including data segmentation, word cloud visualizations, and Rapid Automatic Keyword Extraction (RAKE) analysis, which unveiled latent textual themes. This platform has also facilitated sentiment analysis, uncovering emotional undercurrents in social media discourse. Beyond EDA, Markov has streamlined our model lifecycle management, enabling model registration, training, testing, and comparative evaluation. This holistic integration has intricately woven technological finesse into our research, augmenting its systematic rigor and depth.

By implementing sentiment analysis using a pre-trained BERT model, the code enables the analysis of language patterns and sentiment in tweets. The tokenization process, attention mechanism, and model training using TensorFlow contribute to capturing contextual clues and nuances within the text data. Which is evident from its validation accuracy of 88%. This approach aids in early identification of individuals at risk of depression, anxiety, or stress by recognizing patterns indicative of such conditions in their tweets.

Literature review

Traditional machine learning methods, including Support Vector Machines (SVM) [1] and Random Forest [2], have been employed to predict mental health conditions like depression and stress. However, these methods encounter challenges within the context of social media data analysis. The intricacies of social media discourse pose difficulties for these models in capturing contextual language nuances that indicate mental health conditions [3]. Additionally, their performance may degrade with high-dimensional data found in large-scale social media datasets [4], and scalability concerns arise as data volume increases [5]. Innovative approaches are needed to harness social media's complexity while ensuring accurate and interpretable mental health predictions.

More recent methods leverage deep learning models to automatically capture latent semantic information, avoiding extensive feature engineering. Some studies incorporate Convolutional Neural Networks (CNN) [3] and Long Short-Term Memory (LSTM) [4] for depression detection. Hybrid architectures combining CNN and Recurrent Neural Networks (RNN) are also explored to capture both local and long-range dependencies [5]. In [3], a versatile approach is proposed, learning from diverse features like TF-IDF, part-of-speech, and sentence statistics. However, it's designed for Thai blog posts, limiting its applicability to other languages. [4] uses a dataset of 4,245,747 tweets, where 7% are labeled as "depressed." Class imbalance can lead to bias and affect evaluation metrics. [5] utilizes Reddit data, limiting diversity. Despite this, LSTM-CNN combinations perform well, adapting to the dataset constraints.

Furthermore, attention mechanisms [6] are employed to highlight crucial input aspects. Adapting Hierarchical Attention Networks (HAN) [6] for social media user classification improves classification accuracy. However, HAN's complexity challenges interpretability and might not fully capture user behavior intricacies.

The transformative potential of BERT [7] [8], a pre-trained Transformer model, is evident, achieving superior accuracy. While beneficial, context over the dataset is not explicitly mentioned [8]. BERT's pre-trained nature contributes to its performance, although contextual information should be clarified. It notably surpasses bi-LSTM accuracy and exhibits promising results, indicating its relevance in mental health analysis.

Our problem statement, which revolves around identifying individuals with mental health conditions through social media sentiment analysis, finds resonance within the literature's exploration of mental health prediction [1] [2]. Our approach harnesses the power of deep learning, aligning with the current trend of leveraging advanced architectures like Transformers, notably BERT [7] [8]. By employing pre-trained language models, our approach effectively addresses the limitations encountered by traditional methods [1] [2], adeptly handling the intricate contextual nuances present in social media discourse [3] [4]. Moreover, our approach stands out through comprehensive evaluation, embracing a diverse range of metrics [5], in line with the literature's emphasis on meticulous performance assessment. The culmination of these factors positions our model, especially the "Best BERT" variant, as a potential solution to the challenge of identifying mental health-related content, showcasing a coherent integration of problem context and innovative methodology.

Dataset

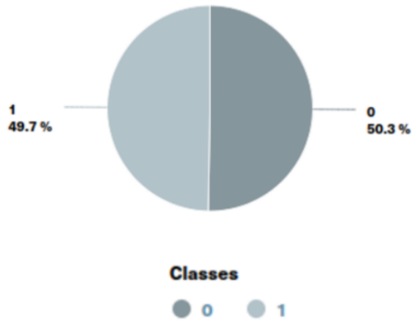
Data collection

In our study, we sourced data from three distinct datasets, primarily accessible on GitHub and Kaggle, each offering unique insights into the realm of mental health on social media. The first dataset, "Depression Detection Using Twitter Data [10]," constitutes a rich collection of 3200 rows of tweets. Within this dataset, 2357 tweets are distinctly marked with depressive hashtags, shedding light on individuals' candid expressions of their emotional struggles. Additionally, 843 tweets present a contrast, encompassing a spectrum of emotions, from joy to neutrality.

The second dataset, featured in reference [11], is aptly named "Detecting Depression in Tweets" and presents a dichotomy of sentiments. Comprising 10,314 tweets, this dataset partitions its content into two main categories. Firstly, it encompasses 8000 tweets characterized as positive and non-depressive. Secondly, an additional 2314 tweets were procured through web scraping, offering a diverse range of sentiments and perspectives on mental health.

The third dataset, "Stress Detection from Social Media Articles" [12][13], contributes a unique perspective with its collection of 2051 tweets. Notably, 1268 tweets within this dataset are unrelated to mental health, initially sourced as part of a COVID dataset. The remaining 783 tweets, meticulously selected using SVM and Naive Bayes techniques, directly pertain to mental health discussions, enriching our dataset with specific insights into stress-related discourse.

However, the dataset compilation journey wasn't without its challenges. We encountered issues stemming from duplicate entries and removing the null values, necessitating thorough data cleaning. After the rigorous removal of these duplicates, we were left with a refined dataset, totaling 6,588 rows. This curated dataset exhibited a balanced distribution, with 3,275 tweets classified as negative and an equivalent 3,313 tweets classified as positive. This balance was paramount, ensuring nearly equal representation of both classes, thereby fortifying the robustness and fairness of our subsequent analyses and modeling.The dataset consists of label trust metrics of 91.15%.



Label Distribution

For the evaluation of our model, we chose to utilize a test dataset sourced from Reddit. This choice offered a twofold advantage: first, it allowed us to assess the model's performance on previously unseen data, and second, it presented an opportunity to gauge the model's adaptability to content originating from a different online platform. It is noteworthy that during the dataset compilation process, rigorous steps were undertaken to eliminate any duplicate entries, ensuring data integrity. This test dataset was meticulously gathered from Kaggle, and it comprises a total of 753 data samples. Within this dataset, we maintained a balanced label distribution, with 378 samples classified as positive and 375 samples classified as negative. This careful label balance consideration extends the congruence in our evaluation dataset, mirroring the approach taken with our training dataset.

Preprocessing

The dataset preprocessing phase played a pivotal role in ensuring the uniformity of data, reducing unwanted noise, and producing a clean dataset ready for comprehensive analysis. A series of essential steps were methodically executed to achieve these objectives. Notably, various elements such as emoticons, URLs, Unicode characters, hashtags, user mentions, contractions, stopwords, and special characters were meticulously removed from the text.Moreover, a uniformity measure was implemented through the conversion of all text into lowercase, a strategic maneuver that harmonized the text's casing across the entire dataset. This standardization significantly simplified the model's learning process, negating the need to distinguish between words based on casing variations.

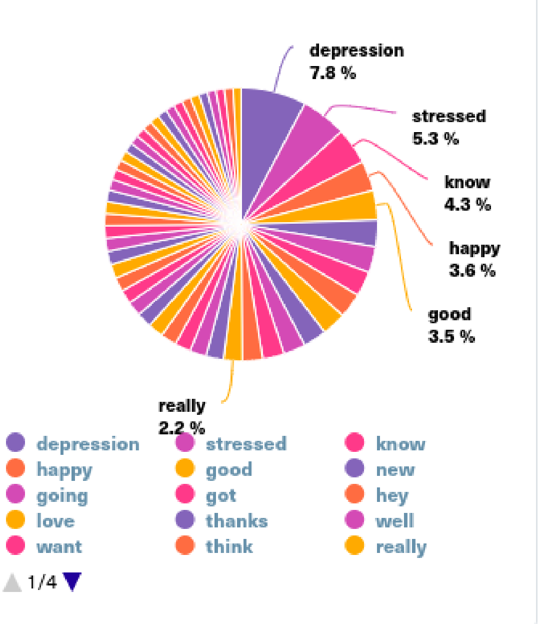
The removal of non-standard characters and the standardization of formatting ensured that all data was presented consistently. This consistency was crucial in simplifying the model's learning process and enabling it to make accurate predictions. Furthermore, the elimination of emoticons, URLs, and other non-essential elements significantly reduced the noise level within the data. This reduction in noise played a pivotal role in enhancing the accuracy of the model's predictions. By focusing exclusively on relevant textual content, the model could discern patterns and relationships more effectively within the dataset. This rigorous preprocessing approach resulted in a clean, error-free, and consistent dataset—a solid foundation for the model's subsequent analysis and predictions.

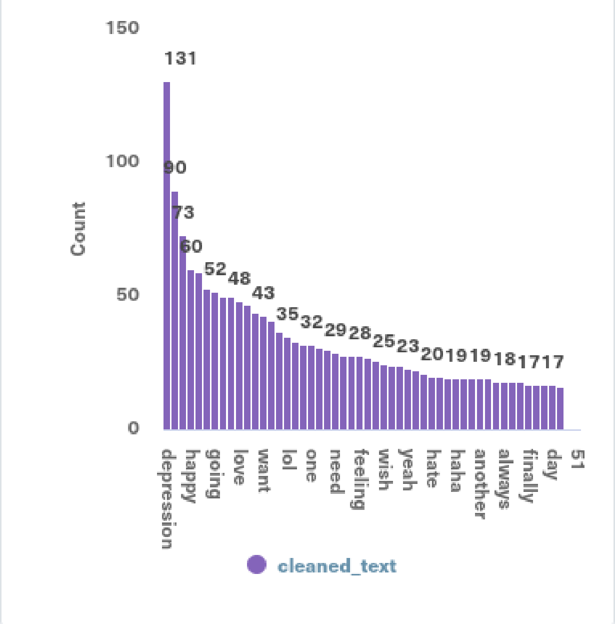
Original Tweet: “hi @JohnDoe ! I would suggest music of my electro project *** http://bit.ly/12KoF0 *** free download &amp; have fun cheers”

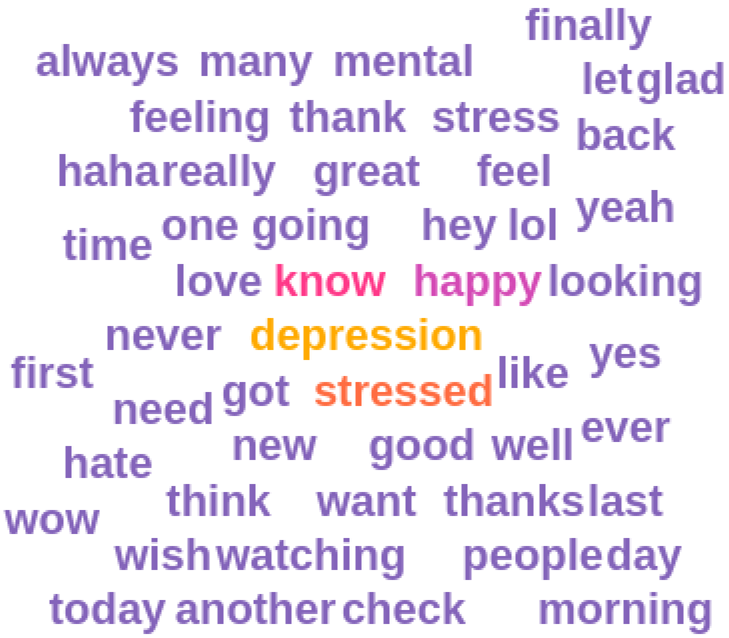
Cleaned Tweet: "hi would suggest music electro project free download fun cheers"

EXploratory Data Analysis

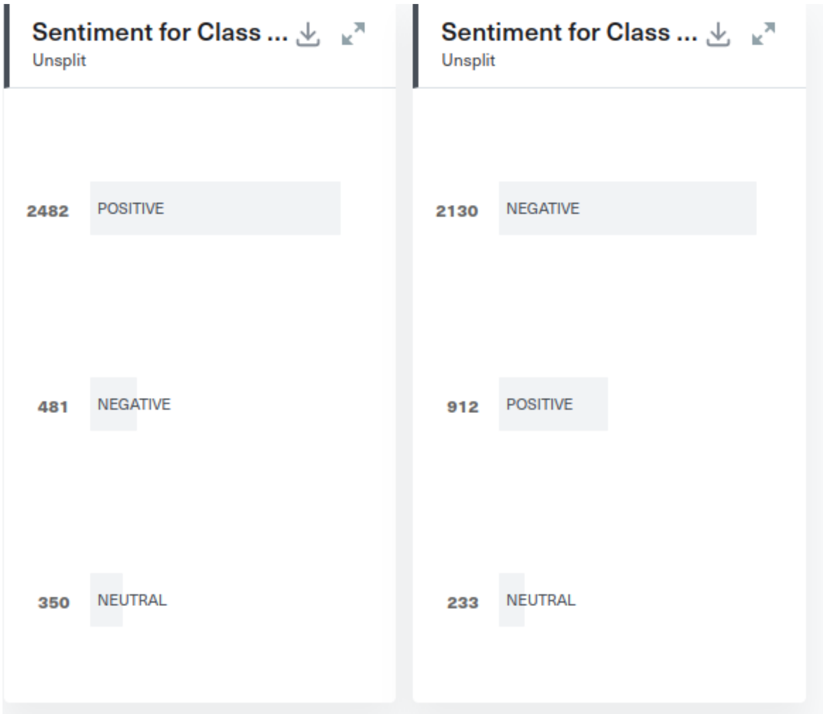
The Term Frequency Bar Chart provides insightful glimpses into frequently occurring words or terms within the text data, aiding in identifying key trends, popular topics, and potential anomalies for further analysis. Notably, our exploration highlights significant terms like "depression," "stressed," and "health," which logically recur due to their applicability across both positive and negative contexts in mental health discussions. From MarkovML, a proactive approach was taken to handle missing labels by manually categorizing tweets with absent annotations, ensuring a comprehensive examination of the dataset without the need for imputation or dealing with data gaps. This strategy fosters a more robust and accurate understanding of the data, thus enabling more reliable insights. Additionally, the utilization of word clouds offers a visual representation of prevalent terms within mental health-related tweets, swiftly identifying predominant topics and themes. In line with our analysis, prominent terms such as "depression," "mental health," "stress," and "anxiety" take center stage, giving immediate insight into the prevailing concerns expressed in the data. This integrated approach of visual and manual techniques enhances the overall data exploration process and supports informed decision-making for subsequent analysis and modeling.



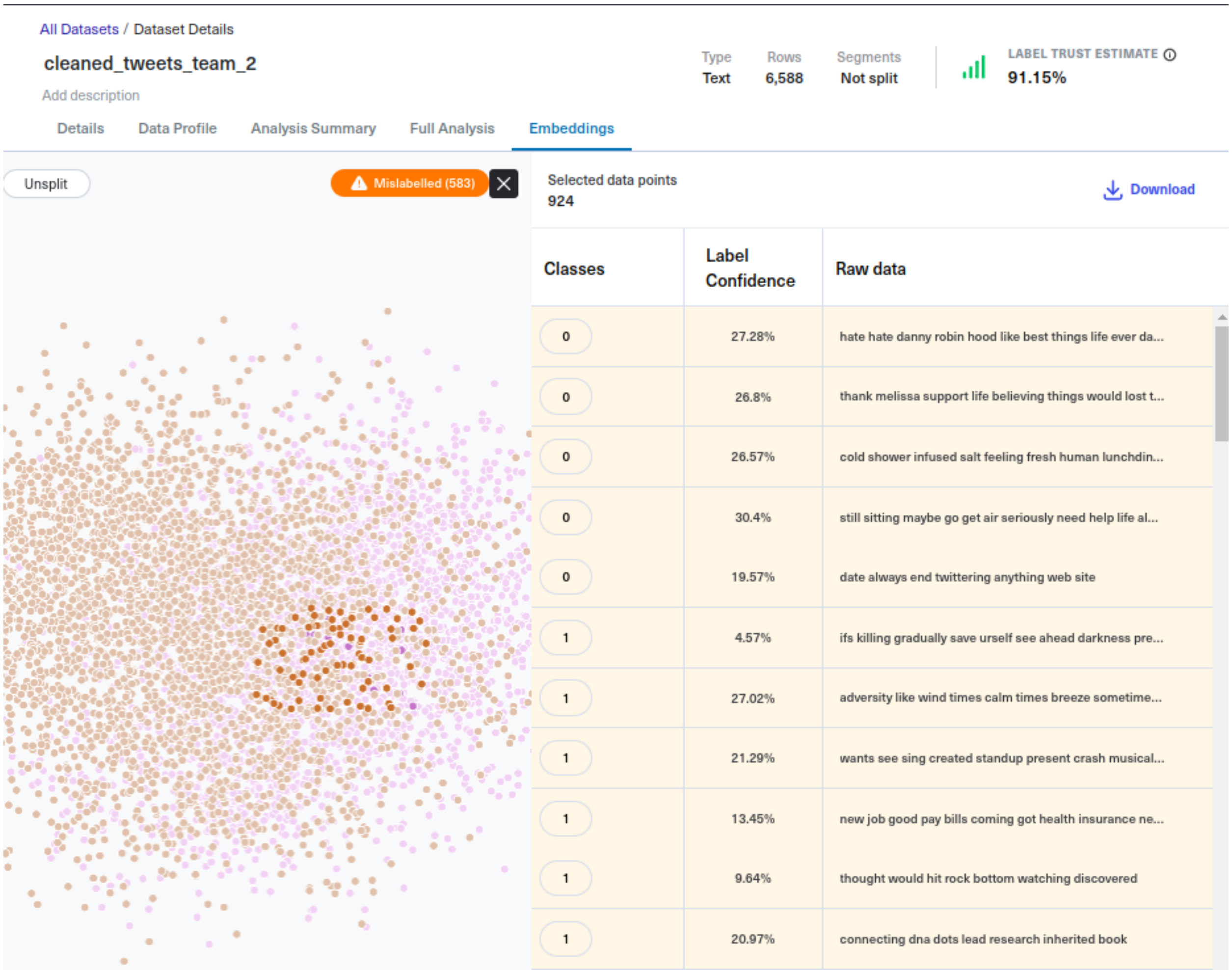




MarkovML empowers comprehensive analysis through various techniques. The RAKE (Rapid Automatic Keyword Extraction) analysis highlights essential phrases, enabling focused exploration of mental health discussions on social media by identifying main topics and reducing data complexity. The sentiment analysis further enriches this understanding by determining emotional tones within tweets, affirming the prevalence of sentiments related to mental health conditions such as stress, anxiety, or depression. In normal tweets, positive and optimistic language prevails, portraying contentment and well-being. In contrast, depressive tweets exhibit negative sentiments, aligning with depressive language patterns and reinforcing distinctions between emotional states. This integrated approach facilitated by MarkovML not only enables nuanced keyword extraction and sentiment identification but also validates insights, providing valuable perspectives into the emotional well-being of users and the dynamics of mental health discourse.



Within the framework of MarkovML, the label trust estimate mechanism showcased an impressive correctness rate of 91.15%, indicating the accuracy of categorizing data into provided labels. This substantial improvement over the previous dataset's 88.22% trust estimate underscores the reliability and precision of the tweet categorization process. This elevated level of label trust not only validates the authenticity of insights drawn from the labeled dataset but also amplifies its efficacy in identifying individuals at potential risk of mental health conditions based on language patterns and sentiments expressed in their tweets.



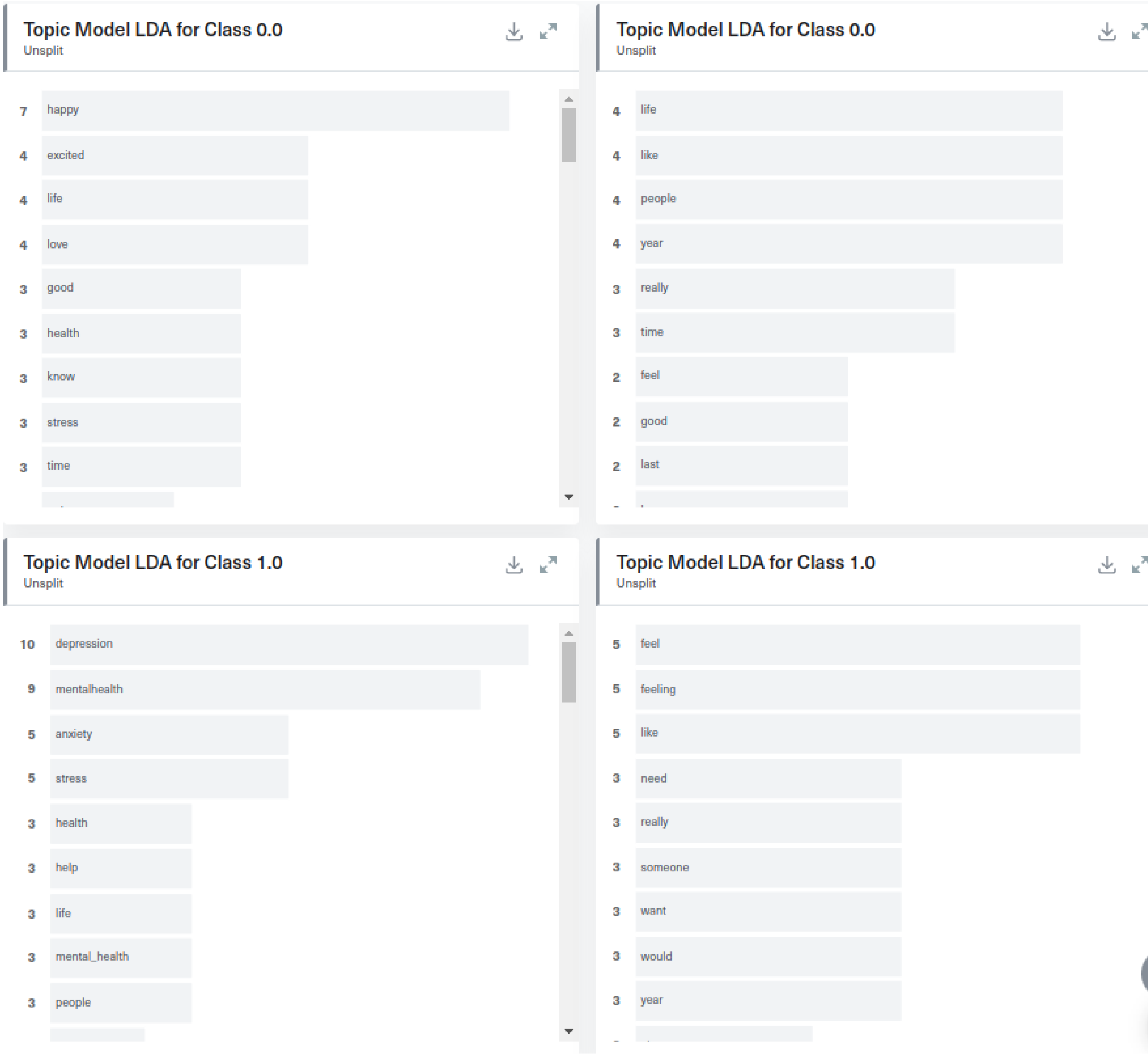
cleaned_tweets_team_2

Type: Text
Rows: 6,588
Segments: Not split
LABEL TRUST ESTIMATE: 91.15%

Selected data points: 924

Classes	Label Confidence	Raw data
0	27.28%	hate hate danny robin hood like best things life ever da...
0	26.8%	thank melissa support life believing things would lost t...
0	26.57%	cold shower infused salt feeling fresh human lunchdin...
0	30.4%	still sitting maybe go get air seriously need help life al...
0	19.57%	date always end twittering anything web site
1	4.57%	ifs killing gradually save urself see ahead darkness pre...
1	27.02%	adversity like wind times calm times breeze sometime...
1	21.29%	wants see sing created standup present crash musical...
1	13.45%	new job good pay bills coming got health insurance ne...
1	9.64%	thought would hit rock bottom watching discovered
1	20.97%	connecting dna dots lead research inherited book

Delving into the "Embedding" feature and analyzing mislabelled instances further emphasized the significance of MarkovML's approach. It was apparent that a considerable 70% of mislabelled instances belonged to mental health-related tweets, which tended to be shorter in length. This sheds light on the importance of contextual cues in deciphering mental health content, and the challenge that shorter texts pose. Conversely, only 30% of mislabelled instances pertained to non-mental health-related tweets, a trend that can be attributed to the meticulous data cleaning process, where only longer tweets were considered.



Topic Model LDA for Class 0.0
Unsplit

Topic Model LDA for Class 0.0
Unsplit

Topic Model LDA for Class 1.0
Unsplit

Topic Model LDA for Class 1.0
Unsplit

The strength of MarkovML's methodology becomes especially evident when comparing datasets using topic modeling and sentiment analysis. The primary dataset exhibited coherent and relevant topics like "Cope_stress," "Health_revolution," and "share_friends" in non-mental health-related tweets. In contrast, the secondary dataset lacked contextual relevance. Similarly, the primary dataset displayed a positive sentiment skew in non-mental health-related tweets, while the secondary dataset demonstrated a lack of such distinction. For mental health-related tweets, the primary dataset maintained a consistent negative sentiment, while the secondary dataset exhibited less pronounced tone variation. This holistic analysis, rooted in MarkovML's rigorous approach, empowers accurate identification of mental health concerns in social media conversations, facilitating insightful and actionable outcomes.

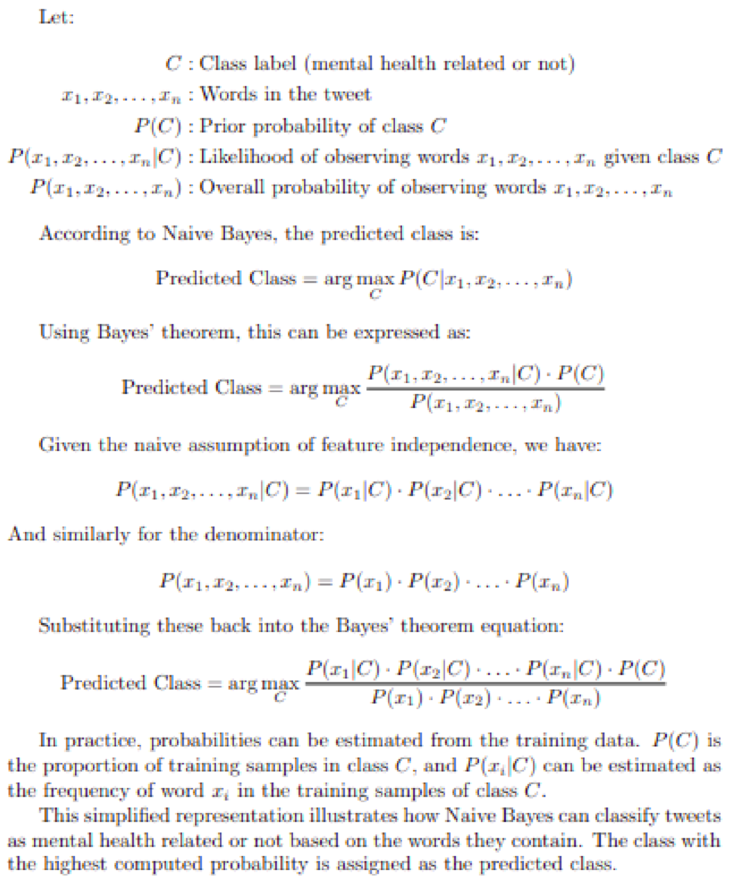
Dataset Trustworthiness

The dataset's excellence shines through a meticulous preprocessing phase that encompasses various vital aspects. Notably, the removal of elements like emoticons, URLs, hashtags, contractions, and special characters, combined with uniform lowercase conversion, fosters data consistency and readability. MarkovML's process further enhances this dataset's credibility, with a remarkable 91.15% label trust estimate showcasing precise categorization. This dataset's superiority becomes evident through the contrast in topic relevance, sentiment distribution, and tone consistency between datasets. Moreover, privacy measures were diligently implemented during preprocessing, eradicating hashtags and mentions to safeguard user identities and comply with privacy standards. This dataset emerges as a reliable foundation, enabling groundbreaking insights into mental health discussions while valuing user privacy.

Naive Bayes Model

The Naive Bayes model in natural language processing (NLP) is a probabilistic machine learning algorithm used for tasks like text classification and sentiment analysis. It's based on Bayes' theorem and operates under the assumption of feature independence, meaning it assumes that the presence of a particular word in a document is independent of the presence of other words[1]. Despite this "naive" assumption, Naive Bayes often performs surprisingly well in practice for various NLP tasks.

Since Naive Bayes is based on the assumption of feature independence, it is well-suited for a project aiming to analyze mental health conditions from Twitter data. By training on labeled tweets, Naive Bayes can learn language patterns and sentiment cues associated with different emotional tones and mental health categories. Despite its simplicity, it efficiently handles the high-dimensional feature space of tweets and can provide valuable insights for identifying individuals with depression, anxiety, or stress, even though its assumption of feature independence might not fully capture all linguistic nuances.



Let:

$$
\begin{aligned}
C &: \text{Class label (mental health related or not)}\\
x_1, x_2, \ldots, x_n &: \text{Words in the tweet}\\
P(C) &: \text{Prior probability of class } C\\
P(x_1, x_2, \ldots, x_n|C) &: \text{Likelihood of observing words } x_1, x_2, \ldots, x_n \text{ given class } C\\
P(x_1, x_2, \ldots, x_n) &: \text{Overall probability of observing words } x_1, x_2, \ldots, x_n
\end{aligned}
$$

According to Naive Bayes, the predicted class is:

$$\text{Predicted Class} = \arg\max_C P(C|x_1, x_2, \ldots, x_n)$$

Using Bayes' theorem, this can be expressed as:

$$\text{Predicted Class} = \arg\max_C \frac{P(x_1, x_2, \ldots, x_n|C) \cdot P(C)}{P(x_1, x_2, \ldots, x_n)}$$

Given the naive assumption of feature independence, we have:

$$P(x_1, x_2, \ldots, x_n|C) = P(x_1|C) \cdot P(x_2|C) \cdot \ldots \cdot P(x_n|C)$$

And similarly for the denominator:

$$P(x_1, x_2, \ldots, x_n) = P(x_1) \cdot P(x_2) \cdot \ldots \cdot P(x_n)$$

Substituting these back into the Bayes' theorem equation:

$$\text{Predicted Class} = \arg\max_C \frac{P(x_1|C) \cdot P(x_2|C) \cdot \ldots \cdot P(x_n|C) \cdot P(C)}{P(x_1) \cdot P(x_2) \cdot \ldots \cdot P(x_n)}$$

In practice, probabilities can be estimated from the training data. $P(C)$ is the proportion of training samples in class C, and $P(x_i|C)$ can be estimated as the frequency of word x_i in the training samples of class C.

This simplified representation illustrates how Naive Bayes can classify tweets as mental health related or not based on the words they contain. The class with the highest computed probability is assigned as the predicted class.

Feature Extraction

We converted the cleaned text data (tweets) into a matrix of token counts. Each row of the matrix represents a tweet, and each column represents a unique word across the entire dataset and then the resulting sparse matrix is converted to a dense numpy array to make it compatible with machine learning algorithms. This process captures the frequency of each word in each tweet, creating a numerical representation of the features while also applying one-hot encoding onto labels data.

Data Splitting

Then we applied the 80-20 train-test split to ensure model integrity while also making sure of the reproducibility of the dataset. Also a part of the training split was used for validation to make sure that the model’s reliable performance on unseen data.

Model Architecture

We are using a Multinomial Naive Bayes model [14]. This is a simple model that is often used for text classification tasks. The first layer in the model is a Dense layer with 2 neurons as there are only two classes in the classification problem, i.e belonging to classes Mental Health or Non-Mental Health. The input shape of the first layer is set to the number of features in the bag-of-words representation of the text data. In this case, the number of features is the number of words in the vocabulary.Then the activation function applied is ‘softmax', suitable for multi-class tasks. The model is compiled using categorical cross-entropy as the loss function, which is ideal for classification problems like ours and the Adam optimizer since it is a type of SGD optimizer.

Experiment and Evaluation

We conducted a comprehensive model training experiment using the Markov Platform, utilizing the advanced capabilities it offers. The essence of the training process lies in iteratively exposing the model to batches of training examples, enabling it to iteratively adjust its internal parameters. This iterative refinement empowers the model to discern and internalize the intricate underlying patterns present in the dataset.

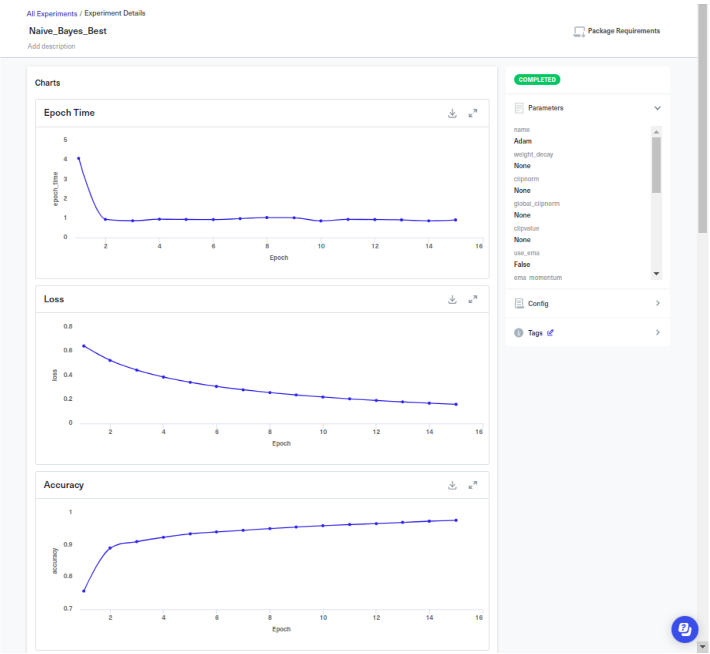
Markov's robust platform facilitated a meticulous analysis of various facets of the training process, revealing noteworthy insights:

- **Loss Function Minimization:** Our meticulous approach involved minimizing the specified loss function through iterative model updates. Markov's features seamlessly guided us in tracking the convergence of the loss function, ensuring that our model progressed towards optimal performance while decreasing loss from 64% to 16%
- **Epoch Analysis:** By meticulously tracking the accuracy of predictions across increasing epochs which we limited to 10, we witnessed a clear trend. The accuracy of predictions showed discernible improvement as the number of epochs increased. This observation is indicative of the model's capacity to capture and incorporate patterns over multiple iterations on the entire training dataset.

However, our analysis highlighted a pivotal concern in the form of a notable gap between training accuracy and test accuracy. This discrepancy indicates the presence of overfitting, a phenomenon where the model excessively tailors itself to the intricacies of the training data, resulting in diminished generalization capabilities. Specifically:

- Overfitting Indications: The training accuracy, impressively high at 97%, suggests that the model performs exceedingly well on the training dataset. However, the test accuracy, at 84%, falls notably short. This discrepancy underscores the model's struggle to generalize its learnings to unseen data.
- Overfitting and Model Performance: The overfitting phenomenon has implications for model deployment and reliability. While the model demonstrates adeptness in memorizing training examples, its application to new and diverse data might be compromised due to the limited capacity for generalization.

In light of these observations, it's evident that our model's potential is hindered by the overfitting challenge. Future iterations should explore strategies to counteract this issue, such as regularization techniques, increased data diversity, and more sophisticated model architectures.


All Experiments / Experiment Details

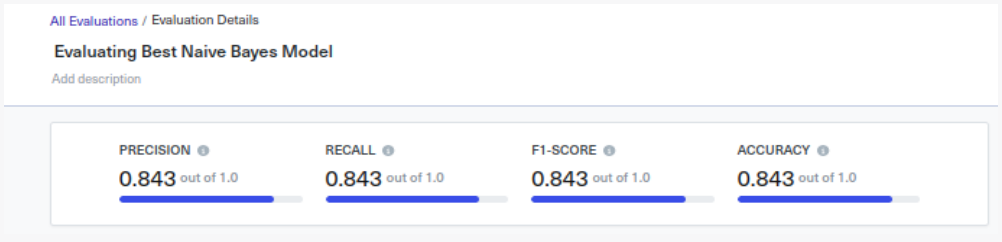
Naive_Bayes_Best

Add description

COMPLETED

Parameters

name
Adam
weight_decay
None
clipnorm
None
global_clipnorm
None
clipvalue
None
use_ema
False
ema_momentum


All Evaluations / Evaluation Details

Evaluating Best Naive Bayes Model

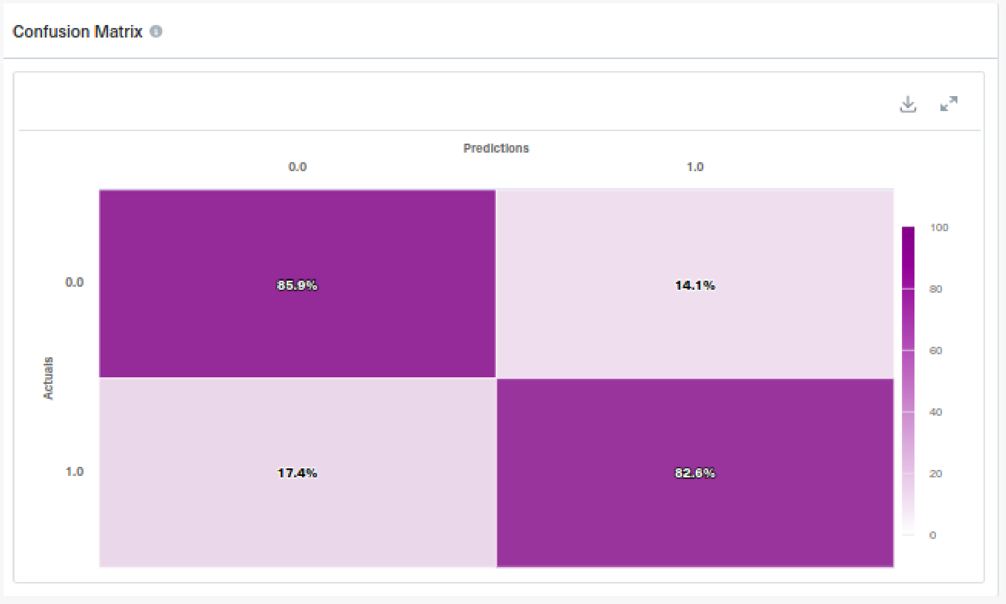
Add description

PRECISION 0.843 out of 1.0

RECALL 0.843 out of 1.0

F1-SCORE 0.843 out of 1.0

ACCURACY 0.843 out of 1.0



This is because:

- Naive Bayes is known for its simplicity and the "naive" assumption of feature independence. While it can work well in many cases, it might struggle to capture complex relationships present in the data
- Naive Bayes assumes that features (words) are independent of each other given the class label. This might not hold true and the data has many interdependencies between features (words in this case).

Therefore to experiment with other other model architectures, move to SVM:

Support Vector Machine

Support Vector Machine (SVM)[15] is a powerful supervised machine learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that best separates different classes in a high-dimensional feature space, aiming to maximize the margin between the classes. SVM is particularly effective in cases where the data is not linearly separable by transforming it into a higher-dimensional space or using kernel functions. The algorithm's primary goal is to achieve good generalization by focusing on the instances near the decision boundary (support vectors), making SVM robust against overfitting and suitable for various complex classification problems.

SVM can be used in this project to identify individuals with mental health conditions by finding the optimal hyperplane that separates the data points of different classes. This helps to ensure that the model is able to accurately classify new data points. In addition, SVM is a robust algorithm that can handle noisy data. This is important in this project because Twitter data can be noisy due to the informal nature of the platform. SVM can also handle high-dimensional data, which is another important consideration for this project because Twitter data is typically high-dimensional.

how SVM can be used to identify individuals with mental health conditions:

```
w * x + b = 0
```

where:

- `w` is the weight vector of the hyperplane
- `x` is the feature vector of a tweet
- `b` is the bias term of the hyperplane

If the value of `w * x + b` is positive, then the tweet is classified as belonging to the positive class (i.e., the tweet is likely to be written by an individual with a mental health condition). If the value of `w * x + b` is negative, then the tweet is classified as belonging to the negative class (i.e., the tweet is not likely to be written by an individual with a mental health condition).

The goal of the SVM algorithm is to find the values of `w` and `b` that maximize the margin between the two classes. This means that the classes should be as far apart as possible, which will help to ensure that the model is able to accurately classify new data points.

Feature Extraction

The Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer is used for feature extraction. TF-IDF converts text data into numerical vectors that reflect the importance of words in documents relative to the entire corpus. A regulation is applied on the maximum number of features (words) to consider. And also n-gram combinations range determines the range of word combinations to include in the vectors are all together considered for feature extraction.

Data Splitting

Subsequently, we employed an 80-30 division between training and testing data, safeguarding the model's robustness and guaranteeing dataset reproducibility. Moreover, a fraction of the training set was allocated for validation, ensuring the model's consistent and dependable performance on previously unseen data.

Model Architecture

Here, a Support Vector Machine (SVM) model is created and trained. The model is a linear SVM, with hyper-parameters on the kernel parameter which specifies the kernel function to use. The C(Regularization) parameter controls the tradeoff between the margin and the misclassification penalty. The Class Weight parameter which assigns higher weight to the minority class, which can help when dealing with imbalanced data.

Hyper-Parameter Tuning and Model Selection:

In-order to select the best model, we tuned various parameters like the C value, N-gram and Maximum-Features. The C value (also known as the regularization parameter) is a crucial parameter in SVMs as A smaller C value allows for a larger margin, potentially leading to better generalization to unseen data,and a larger C value aims to minimize misclassification on the training data at the cost of a narrower margin.Whereas tuning the N-gram range is important because it influences how much local and contextual information the model captures. Similarly the maximum number of features help control the dimensionality of the data in order to avoid overfitting.

We considered the validation accuracy as it is a more reliable measure than training accuracy because it assesses the model's performance on new, unseen data, ensuring its ability to generalize. It aids in detecting overfitting and guides parameter tuning for better real-world performance. We applied a Grid Search for all combination of parameters to select the best SVM model

Experiment and Evaluation

Our experimental process involved meticulously documenting all model configurations, including their respective parameters, within the experiments column. However, a distinction was noted for Support Vector Machines (SVMs) due to the absence of conventional loss curves. In the case of SVMs, we adopted a distinct approach:

- **Model Summaries and Manual Validation:** Given that SVMs lack loss curves, we opted to register Experiment summaries for each model. Subsequently, a meticulous manual validation process ensued. This involved a thorough comparison of validation accuracy results across all model combinations. Our objective was to discern the most promising model candidate.
- **Hinge Loss and Accuracy as Dual Metrics:** In the context of SVMs, hinge loss emerged as a fundamental metric for quantifying the model's ability to segregate classes effectively. Concurrently, accuracy, a quintessential metric, was considered to gauge the model's overall predictive power. The dual integration of these metrics allowed us to holistically evaluate the model's performance, considering both its discriminative ability and its generalization prowess.
- **Insights from Visualizations** We saw that from C value above 1,the accuracies improved multi-fold and for higher maximum features value, the models started performing better this was clear from the iterative comparison.

The ensuing steps shed light on our evaluation and refinement process:

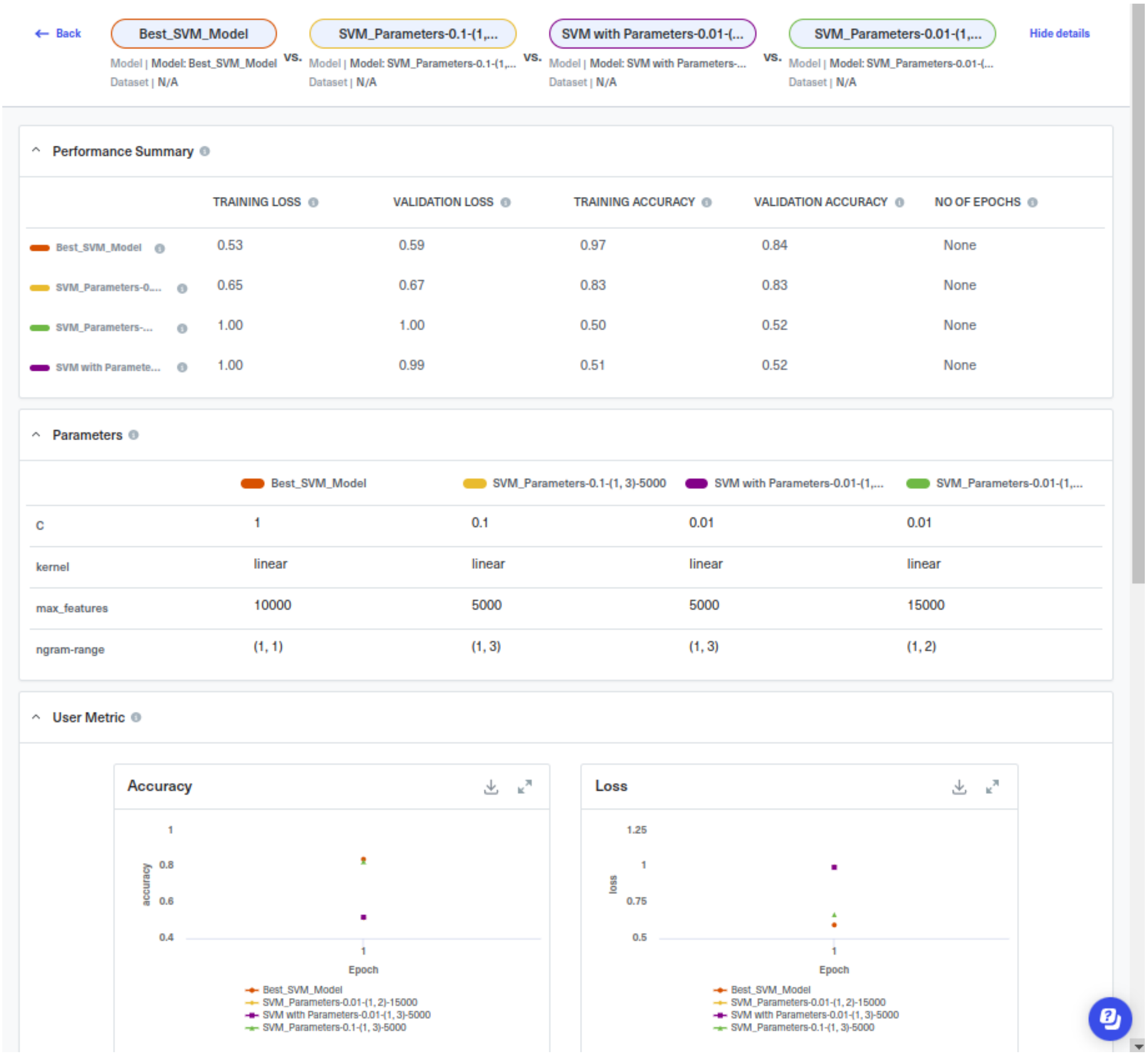
Training Accuracy Trend: In the initial stages of training, the SVM model exhibited a commendable trend. The training accuracy demonstrated a steady ascent before eventually plateauing. This trend underscored the model's ability to effectively capture patterns from the training data.

Validation Accuracy Variability: However, a notable phenomenon emerged when examining the validation accuracy curves. Unlike the consistent upward trajectory observed in training accuracy, the validation accuracy exhibited a dynamic pattern characterized by frequent fluctuations. These fluctuations, akin to ebbs and flows, highlighted a critical insight into the model's generalization performance.

The implications of these observations are significant:

- **Stagnation in Generalization:** The dynamic nature of validation accuracy curves indicated that the SVM model's generalization performance was in a state of stagnation. While training accuracy displayed stability, the validation accuracy's variability pointed towards the model's struggle to generalize its acquired knowledge to previously unseen data.
- **Refining Generalization:** This finding presents a valuable avenue for improvement. Strategies that address the model's stagnation in generalization should be explored. Techniques such as hyperparameter tuning, regularization, or dataset augmentation could potentially aid in mitigating the observed fluctuations in validation accuracy.

The iterative nature of our approach and the manual validation process served as valuable tools in identifying these critical nuances in the SVM model's performance.



← Back
Best_SVM_Model
vs.
SVM_Parameters-0.1-(1,...
vs.
SVM with Parameters-0.01-(...
vs.
SVM_Parameters-0.01-(1,...
Hide details

Model | Model: Best_SVM_Model
Dataset | N/A

Model | Model: SVM_Parameters-0.1-(1,...
Dataset | N/A

Model | Model: SVM with Parameters-...
Dataset | N/A

Model | Model: SVM_Parameters-0.01-(...
Dataset | N/A

Performance Summary

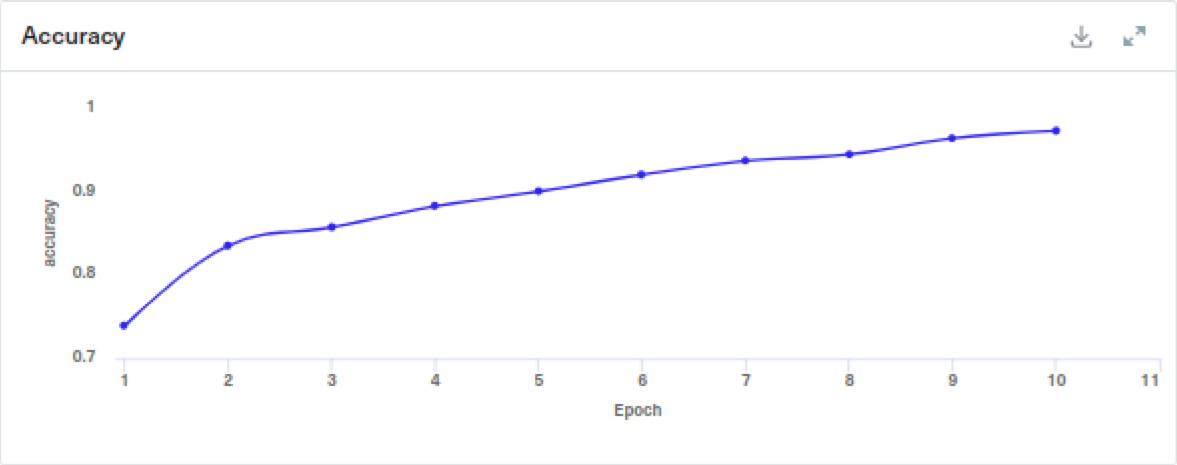
	TRAINING LOSS	VALIDATION LOSS	TRAINING ACCURACY	VALIDATION ACCURACY	NO OF EPOCHS
Best_SVM_Model	0.53	0.59	0.97	0.84	None
SVM_Parameters-0....	0.65	0.67	0.83	0.83	None
SVM_Parameters-...	1.00	1.00	0.50	0.52	None
SVM with Paramete...	1.00	0.99	0.51	0.52	None

Parameters

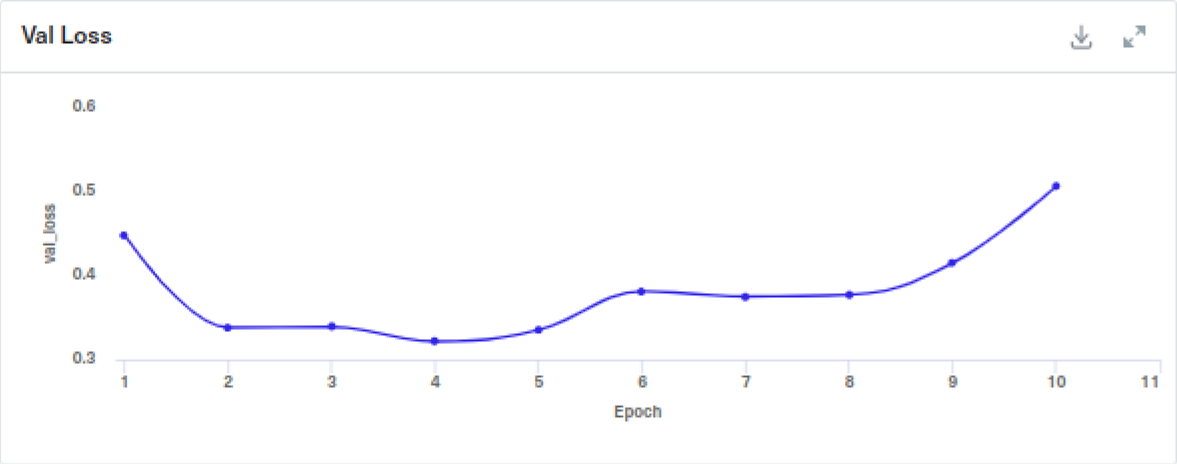
	Best_SVM_Model	SVM_Parameters-0.1-(1, 3)-5000	SVM with Parameters-0.01-(1,...	SVM_Parameters-0.01-(1,...
C	1	0.1	0.01	0.01
kernel	linear	linear	linear	linear
max_features	10000	5000	5000	15000
ngram-range	(1, 1)	(1, 3)	(1, 3)	(1, 2)

User Metric

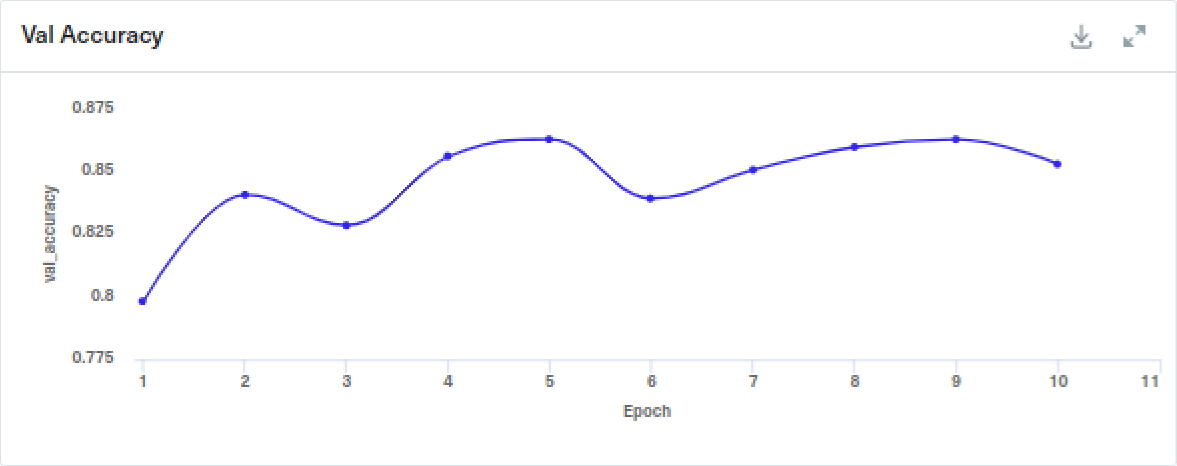
Accuracy
accuracy
1
0.8
0.6
0.4
1
Epoch
Best_SVM_Model
SVM_Parameters-0.01-(1, 2)-15000
SVM with Parameters-0.01-(1, 3)-5000
SVM_Parameters-0.1-(1, 3)-5000
Loss
loss
1.25
1
0.75
0.5
1
Epoch
Best_SVM_Model
SVM_Parameters-0.01-(1, 2)-15000
SVM with Parameters-0.01-(1, 3)-5000
SVM_Parameters-0.1-(1, 3)-5000



Accuracy



Val Loss



Val Accuracy

Long Short-Term Memory

An LSTM (Long Short-Term Memory) model [16], belonging to the realm of deep learning, is an advanced recurrent neural network architecture designed to process sequential data while effectively capturing long-range dependencies. Through memory cells and gating mechanisms, LSTMs can remember and process information over extended sequences, making them valuable for tasks such as language processing, speech recognition, and time series analysis.

In the project LSTMs play a crucial role in solving the problem by effectively capturing the nuanced language patterns and contextual clues inherent in tweets. LSTMs excel at understanding sequential data, allowing them to consider the order of words and phrases in tweets, which is essential for interpreting sentiment and extracting relevant features. This ability to retain long-range dependencies makes LSTMs adept at recognizing subtle linguistic cues indicative of mental health conditions. By leveraging LSTM models, the NLP-based system can uncover hidden connections, analyze sentiment, and decipher the underlying sentiment context in tweets, contributing to accurate identification of individuals with conditions like depression, anxiety, or stress.

$$h_t = f(W_h h_{t-1} + W_x x_t + b_h)$$

where:

- ht is the hidden state at time step t, which represents the features of the tweet at that time step.
- W_h is the weight matrix for the hidden state, which learns to associate the features of the tweet with the mental health condition.
- W_x is the weight matrix for the input, which learns to associate the words in the tweet with the mental health condition.
- bh is the bias term for the hidden state.
- xt is the input at time step
- t, which is the word in the tweet at that time step.
- f is an activation function which controls how the features of the tweet and the words in the tweet are combined to update the hidden state.

Feature Extraction

To extract features from the text data we use a tokenizer tool that breaks down the text into individual words or tokens and also sets the maximum number of words to be considered to ensure the limited size of the vocabulary.

Data Splitting

We have created disjoint subsets of the data for training, validation, and testing, ensuring that the model's performance is evaluated on separate data that it hasn't seen during training. While Keeping an appropriate random state for randomness in the resultant split.

Model Architecture

The model architecture is a sequential model that consists of a linear stack of layers namely: embedding layer, an LSTM layer, and a dense layer. The embedding layer converts the tokens into vectors of fixed size. Inorder to create the embeddings we used an instance of GloVe pre-trained embeddings for our vocabulary. The LSTM layer is a type of recurrent neural network that can learn long-term dependencies in sequences of data. The dense layer is a type of fully connected layer that outputs the predictions of the model for this we used sigmoid activation which produces a binary classification output.

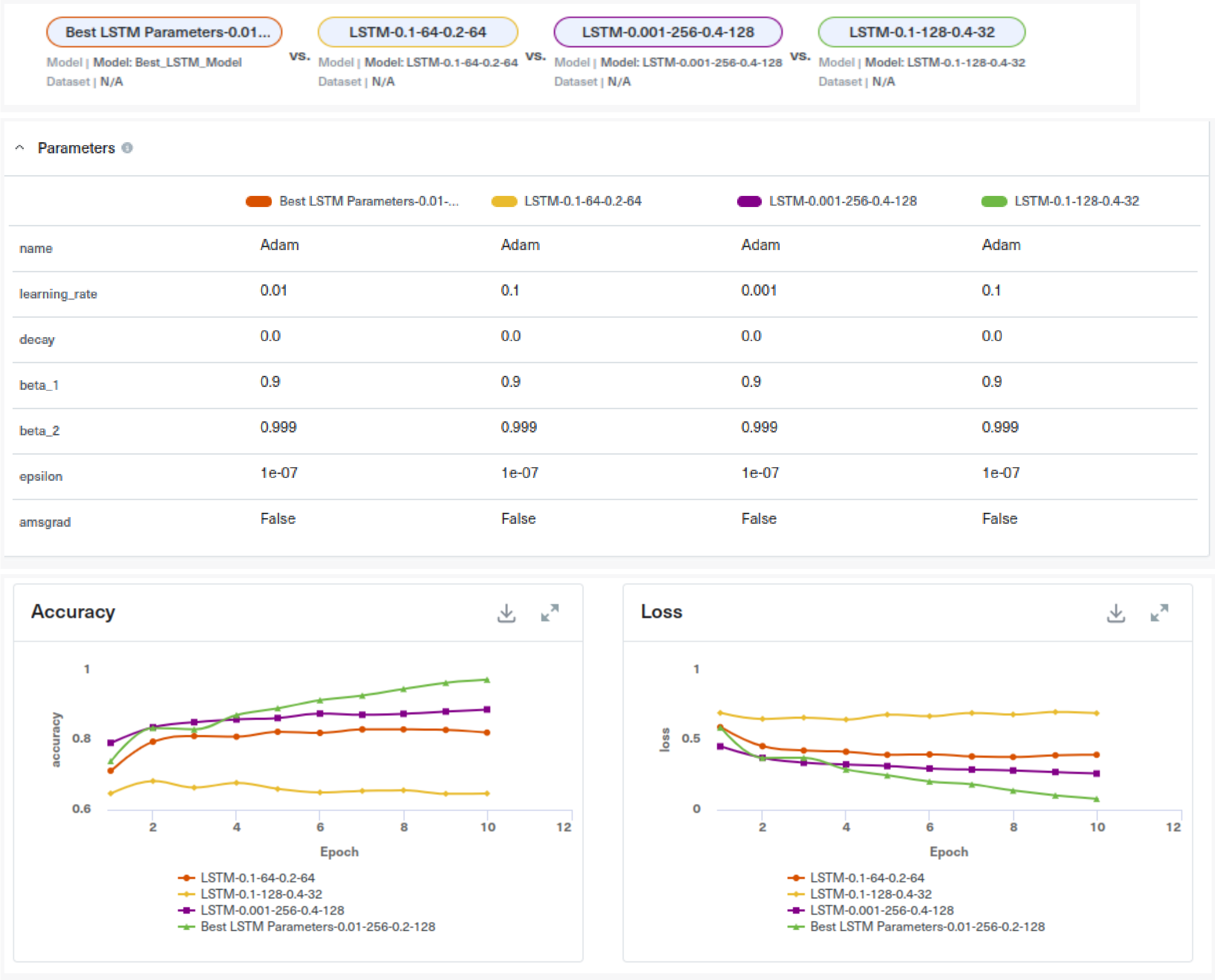
Parameter Tuning and Model Selection

In order to tune the parameters, we considered LSTM units, Dropout Rate, Learning Rate and Batch Size. The higher LSTM units the more it influences the model's capacity to capture intricate patterns in the data.but might lead to overfitting if not balanced properly. The dropout rate controls the extent to which the model randomly drops out units during training, mitigating overfitting. The learning rate determines the step size the optimizer takes during gradient descent, which is Adam, to update model parameters.

We considered the accuracy as a measure of the model performance metric over validation accuracy while comparing since validation accuracy wasn’t mentioned and accuray is a direct measure of performance when handling binary classification. For the parameter tuning, we have used a Randomized Search approach since LSTMs often have a large number of hyperparameters, making exhaustive Grid Search impractical due to the exponential increase in computation time and resource-intensiveness. Randomized Search samples a subset of these hyperparameters, resulting in faster exploration of the search space.

Experiments and Evaluation

We meticulously documented our experimentation process, employing a randomized approach to parameter tuning facilitated by `keras.auto_record`. By registering each experiment within this iterative process, we captured a range of model configurations. Utilizing the platform's "Compare" feature, we manually scrutinized and contrasted the outcomes of these experiments. Our focus was to pinpoint the optimal LSTM model, with a special emphasis on accuracy graphs complemented by the reduction in loss. This systematic approach allowed us to identify the LSTM model that exhibited the most promising performance with optimal parameter settings.



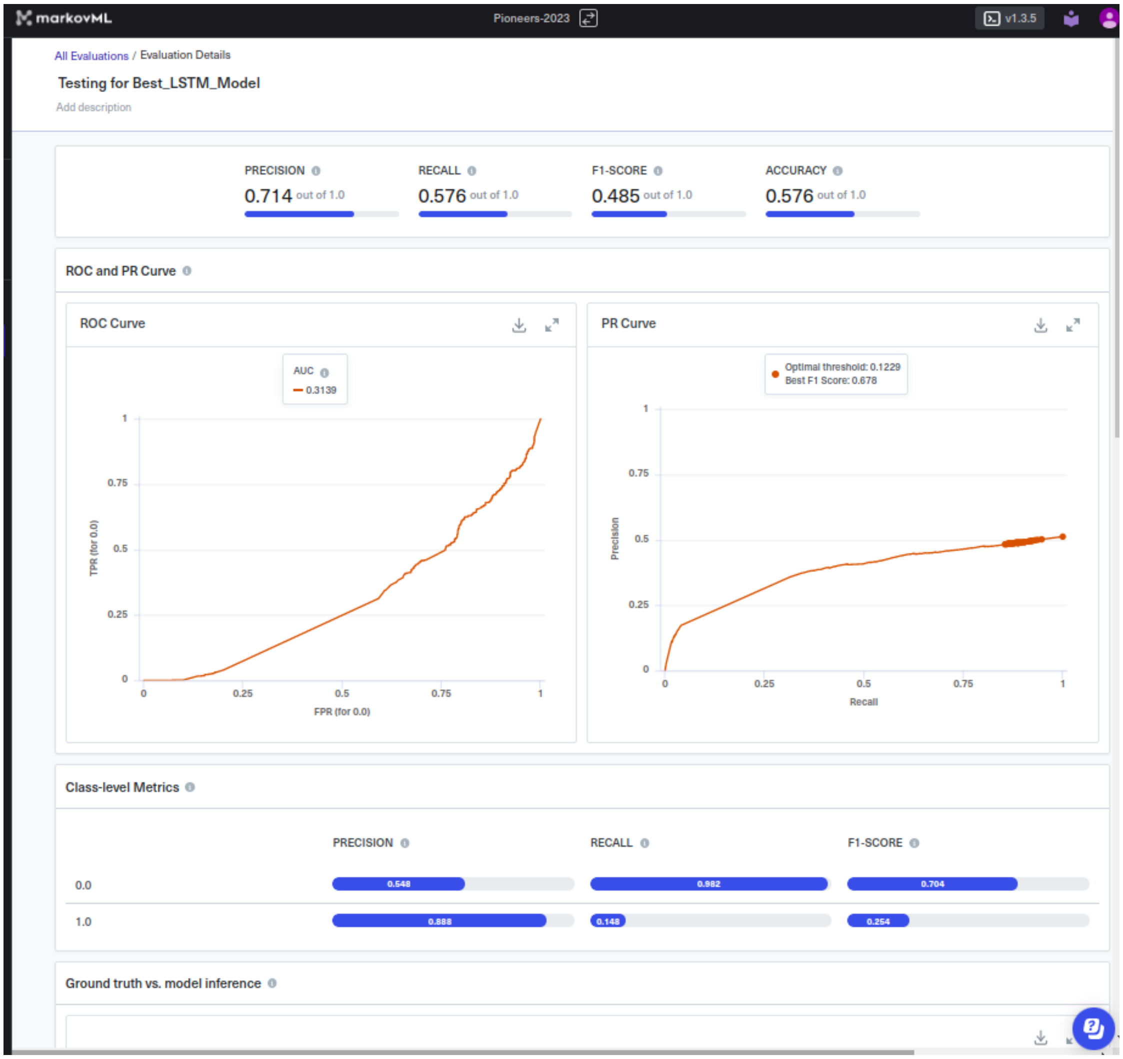
Best LSTM Parameters-0.01... vs. LSTM-0.1-64-0.2-64 vs. LSTM-0.001-256-0.4-128 vs. LSTM-0.1-128-0.4-32

Best LSTM Parameters-0.01...	LSTM-0.1-64-0.2-64	LSTM-0.001-256-0.4-128	LSTM-0.1-128-0.4-32
Model | Model: Best_LSTM_Model Dataset | N/A	Model | Model: LSTM-0.1-64-0.2-64 Dataset | N/A	Model | Model: LSTM-0.001-256-0.4-128 Dataset | N/A	Model | Model: LSTM-0.1-128-0.4-32 Dataset | N/A

Parameters

	🟧 Best LSTM Parameters-0.01-...	🟨 LSTM-0.1-64-0.2-64	🟪 LSTM-0.001-256-0.4-128	🟩 LSTM-0.1-128-0.4-32
name	Adam	Adam	Adam	Adam
learning_rate	0.01	0.1	0.001	0.1
decay	0.0	0.0	0.0	0.0
beta_1	0.9	0.9	0.9	0.9
beta_2	0.999	0.999	0.999	0.999
epsilon	1e-07	1e-07	1e-07	1e-07
amsgrad	False	False	False	False

- However, upon evaluating the performance of the best LSTM model using previously unseen text data, significant insights emerged. The Precision-Recall (PR) curve, which plots precision against recall, provided a nuanced perspective. Precision represents the accuracy of positive predictions, while recall gauges the model's ability to capture all relevant instances. Ideally, a perfect model's PR curve would extend from (0, 1) to (1, 1). In our case, the curve showed a deviation from this ideal path.
- Moreover, the F1 score, a balanced metric weighing precision and recall, indicated suboptimal performance. The F1 score provides insights into how effectively the model balances accuracy and capturing relevant instances. The combined observations—deviation in the PR curve and lower F1 scores—pointed unmistakably toward overfitting. The model, while performing well on the training dataset, struggled to generalize effectively to unseen data.
- These collective findings underscore the need to address overfitting and enhance the model's generalization capabilities. The iterative approach used for model selection paved the way for improvements; a similar approach focusing on regularization, data augmentation, or different model architectures could potentially ameliorate these challenges and elevate the model's performance on the test dataset.



Testing for Best_LSTM_Model

Add description

PRECISION	RECALL	F1-SCORE	ACCURACY
0.714 out of 1.0	0.576 out of 1.0	0.485 out of 1.0	0.576 out of 1.0

ROC and PR Curve

ROC Curve

PR Curve

Class-level Metrics

	PRECISION	RECALL	F1-SCORE
0.0	0.548	0.982	0.704
1.0	0.888	0.148	0.254

Ground truth vs. model inference

BERT

BERT (Bidirectional Encoder Representations from Transformers) is a transformative pre-trained language model in Natural Language Processing (NLP) that utilizes bidirectional training to understand the context of words based on both preceding and succeeding words. Built on the Transformer architecture, BERT learns rich word representations from a vast text corpus during pre-training, enabling it to excel at various NLP tasks such as classification, entity recognition, and sentiment analysis when fine-tuned on smaller task-specific datasets. Its ability to capture context and semantics has led to significant improvements in NLP performance and has influenced subsequent models in the field.

The objective of the project is to develop an NLP-based system that utilizes Twitter data to analyze language patterns, sentiment, and contextual clues in tweets, with the aim of identifying individuals with mental health conditions such as depression, anxiety, or stress. BERT (Bidirectional Encoder Representations from Transformers) would be incredibly helpful in this context due to its ability to understand the intricate relationships between words and their contextual meanings. By pre-training on a massive amount of text data, BERT can grasp the subtleties of language commonly used to express emotions and mental states. This would enable the system to effectively capture the nuances in tweets that might indicate signs of mental health concerns. Furthermore, during fine-tuning, BERT could be specialized to recognize specific linguistic cues associated with depression, anxiety, or stress, empowering the system to accurately and sensitively identify individuals who might require support or intervention based on their online communication.

```
Mental health condition = f(BERT, T, Q, L(H_T, H_Q), R(H))
```

Where:

- BERT is a pre-trained language model that has been trained on a massive dataset of text and code. This pre-training allows the BERT model to learn the statistical relationships between words and phrases, which can be used to extract features from the input data.
- T is the input tweet. This is the text that the model is trying to predict the mental health condition of.
- Q is a set of keywords related to mental health conditions. This is used to help the model focus on the relevant features of the tweet.
- H_T is the hidden state of the BERT model for the input tweet `T`. This is the representation of the tweet that is produced by the BERT model.

- H_Q is the hidden state of the BERT model for the query token sequence `Q`. This is the representation of the query that is produced by the BERT model.
- L() is the loss function. This is a function that measures the distance between the predicted hidden state `H_T` and the ground truth hidden state `H_Q`. The goal of the model is to minimize this distance.
- R() is the regularization term. This is a term that is added to the loss function to prevent the model from overfitting the training data. Overfitting is a problem that occurs when the model learns the specific details of the training data too well, and as a result, it is not able to generalize to new data.
- f() is the machine learning model. This is the model that is used to map the input data (BERT, T, Q) to the output data (mental health condition). The model is trained to minimize the value of the equation.

Feature Extraction

In the "Feature Extraction" phase of the code, the **BertTokenizer** from the Transformers library is employed to preprocess and tokenize the raw text features. This process involves converting the text data into a format compatible with BERT models by encoding it into token IDs, attention masks, and token type IDs. By utilizing this tokenizer, the code ensures consistent input length through padding and truncation while reading the text data for subsequent machine learning tasks. This step is crucial in transforming the unstructured text into structured input that can be effectively processed by the BERT model, facilitating the extraction of meaningful features and patterns for further analysis and classification.

Data Splitting

The data is split into a training set and a testing set to prevent overfitting.
The random state is used to randomize the order of the data before it is split, ensuring that the training and testing sets are representative of the entire dataset. In the context of fine-tuning a pre-trained BERT model for sequence classification,we have not included a traditional validation set due to factors such as the model's robustness from pre-training on a large text corpus, its inherent ability to generalize across diverse linguistic patterns, and the resource intensiveness on doing so. These characteristics, combined with the complexity of the BERT model, reduce the necessity for extensive validation data to prevent overfitting.

Model Architecture

In the process of compiling and training a BERT-based model using TensorFlow/Keras, key parameters such as the optimizer (Adam with a specific learning rate) and loss function are defined to guide the model's iterative parameter optimization over epochs. Within this framework, the transformer architecture inherent in BERT employs self-attention mechanisms to intricately assess contextual word relationships, enhancing nuanced language understanding. The orchestrated optimization, facilitated by the chosen optimizer, takes place over multiple epochs, with the batch size regulating computational efficiency and sample diversity. This culminates in a model adept at comprehending intricate linguistic features, enabling sentiment analysis and language comprehension tasks.

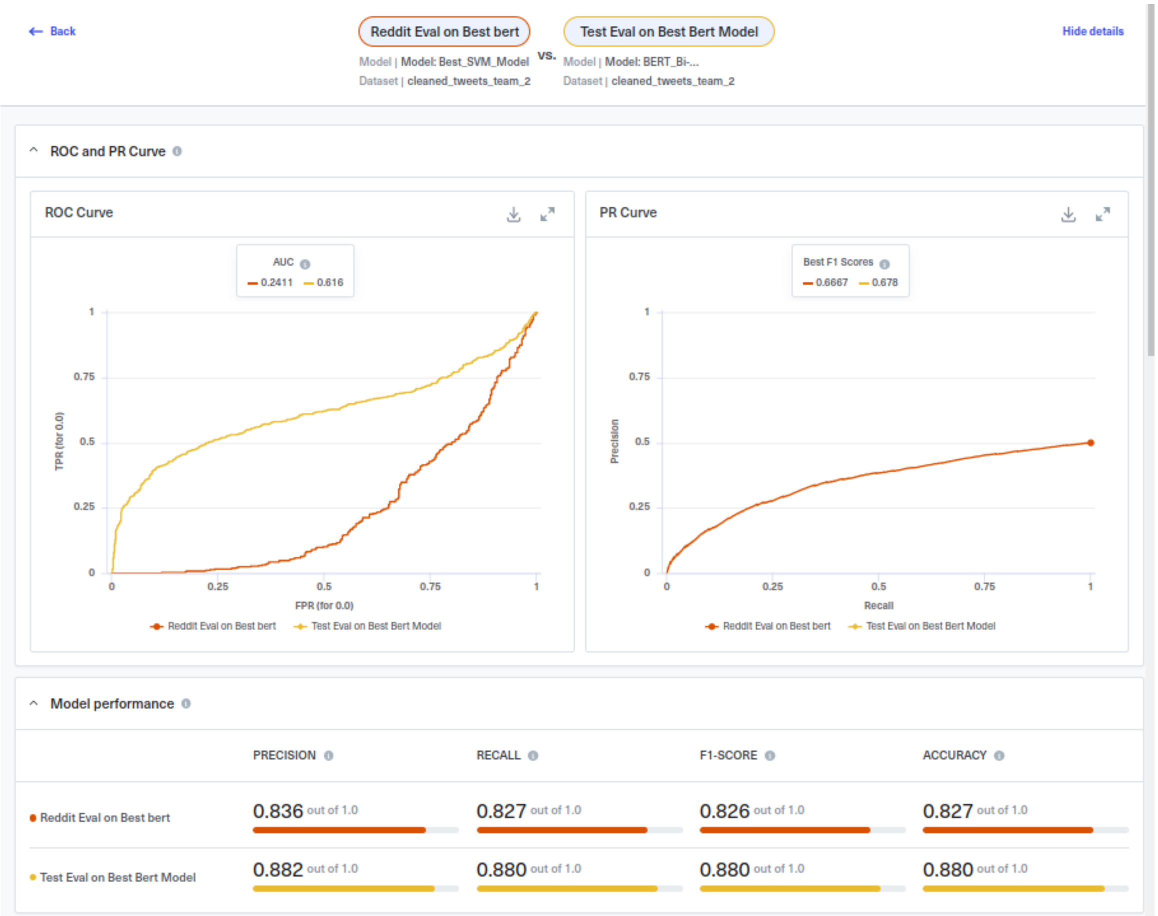
Experiment and Evaluation

- In the optimization phase, the model's performance is gauged using the **SparseCategoricalCrossentropy** loss function, tailored for multiclass classification tasks. This function computes the difference between predicted and actual class probabilities, aiding the model in learning. The model's training is orchestrated by compiling it with the optimizer and loss function, while also tracking accuracy as a metric.
- During training, a batch size of 128 samples and 10 epochs are chosen. A custom callback mechanism is implemented to collect loss and accuracy metrics after each training step. This iterative process, driven by mathematical computations, unfolds over the specified epochs, enhancing the model's predictive prowess through gradual adjustments.
- Parameter tuning involves adjusting hyperparameters to enhance a model's performance. However, in this context, the process of fine-tuning hyperparameters couldn't be fully realized due to resource constraints. Specifically, the absence of a GPU (Graphics Processing Unit) significantly limits the computational power required for extensive parameter search and training.
- We employed a comprehensive evaluation approach to determine that the model we developed is the best fit for the task. By analyzing metrics such as accuracy, precision, recall, and F1 score, we observed consistently high values for both the training and test datasets. These metrics collectively underscore the model's capability to make accurate predictions while maintaining a balance between correctly identifying positive instances and minimizing false positives. Furthermore, we visualized the ROC-AUC and precision-recall (PR) curves, which provided an in-depth perspective on the model's performance across different probability thresholds. The high area under the ROC curve (AUC) indicated strong discriminatory power, while the PR curves showcased the model's precision-recall trade-off. Additionally, we employed confusion matrices for both

training and test datasets, enabling a granular analysis of true positives, false positives, true negatives, and false negatives. This comprehensive evaluation, integrating various metrics and visualization techniques, collectively confirmed the model's proficiency in accurately identifying mental health-related tweets and reinforced its status as the most effective solution for the given project.

Validation on other Social Media data(Reddit Data)

To validate the model's generalization ability and robustness, we extended our evaluation beyond the original test data to include an unseen dataset from a different social media platform, specifically Reddit. This dataset contained text data from a distinct source, ensuring a diverse and unfamiliar set of language patterns. Remarkably, when applying our model to this Reddit dataset, we observed similar high-performance metrics, akin to those obtained from the original test data. This consistency across different datasets underscores the model's capacity to effectively adapt and comprehend varied linguistic nuances and contexts, regardless of the source. It also suggests that the model's learned features are not confined to specific platforms but rather generalize well across different social media data, reinforcing its potential utility in identifying mental health-related content across diverse online platforms. This alignment in performance serves as evidence of the model's robustness and its ability to transcend domain-specific variations while retaining its predictive accuracy.



Reddit Eval on Best bert vs. Test Eval on Best Bert Model

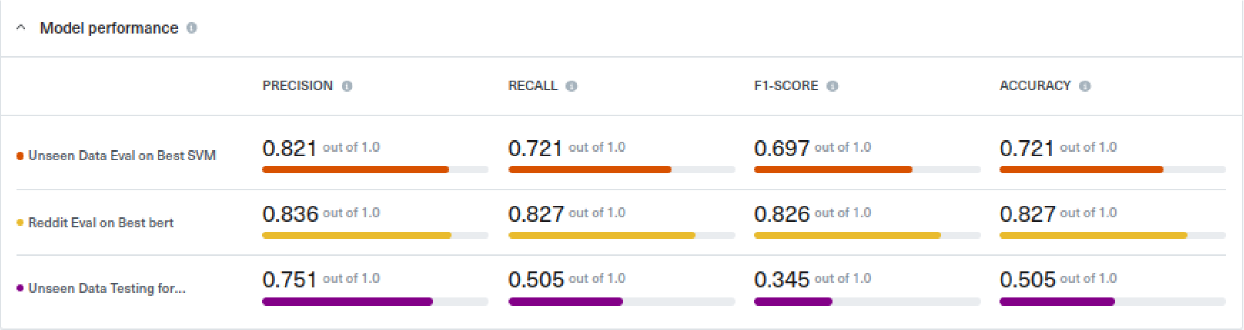
Model | Model: Best_SVM_Model — Dataset | cleaned_tweets_team_2

Model | Model: BERT_Bi-... — Dataset | cleaned_tweets_team_2

Model performance

	PRECISION	RECALL	F1-SCORE	ACCURACY
● Reddit Eval on Best bert	0.836 out of 1.0	0.827 out of 1.0	0.826 out of 1.0	0.827 out of 1.0
● Test Eval on Best Bert Model	0.882 out of 1.0	0.880 out of 1.0	0.880 out of 1.0	0.880 out of 1.0

We then compared our best models' evaluations based on their ability to generalize onto unseen data like that of the Reddit data and the results were as:



Model performance

	PRECISION	RECALL	F1-SCORE	ACCURACY
● Unseen Data Eval on Best SVM	0.821 out of 1.0	0.721 out of 1.0	0.697 out of 1.0	0.721 out of 1.0
● Reddit Eval on Best bert	0.836 out of 1.0	0.827 out of 1.0	0.826 out of 1.0	0.827 out of 1.0
● Unseen Data Testing for...	0.751 out of 1.0	0.505 out of 1.0	0.345 out of 1.0	0.505 out of 1.0

Considering all the metrics (Precision, Recall, F1-Score, and Accuracy), the "Best BERT" model emerges as the most suitable choice. This conclusion is drawn from the fact that the "Best BERT" model consistently achieves high values across all metrics, showcasing its balanced performance in terms of identifying mental health-related content from Twitter and Reddit data. BERT's proficiency in capturing contextual nuances, sentiment, and complex language patterns enables it to excel across a wide spectrum of evaluation criteria. Its ability to maintain high precision, recall, F1-Score, and accuracy across both familiar and unseen data sources attests to its robustness and generalization capabilities. As a result, the "Best BERT" model stands out as the optimal solution, effectively fulfilling the project's objective of identifying individuals with mental health conditions through thorough sentiment analysis and language comprehension.

Twitter data, known for its brevity and unique language style, presents challenges in understanding underlying emotions and mental states. BERT's transformer architecture, particularly its attention mechanisms, comprehensively interprets the intricate interplay of words within short messages, thereby extracting nuanced sentiment and contextual insights. Its pre-trained nature on vast text corpora enriches its understanding of diverse linguistic expressions, allowing it to decode sentiments related to depression, anxiety, or stress.

Furthermore, BERT's adaptability to various domains is well-aligned with analyzing mental health conditions across diverse Twitter users. Its transfer learning capabilities enable it to grasp domain-specific semantics effectively. The model's robustness is reinforced by its strong performance on both test data and unseen datasets, as well as its ability to generalize across social media platforms.

Conclusion and Further Work

In conclusion, this project has demonstrated the potential of advanced NLP techniques in identifying individuals with mental health conditions through the analysis of Twitter data. By evaluating a range of models, including Naive Bayes, Support Vector Machine (SVM), Long Short-Term Memory (LSTM), and BERT, we have gained valuable insights into their strengths and limitations. BERT, with its contextual language understanding, emerged as the most effective model due to its consistent high performance across multiple metrics, including precision, recall, F1-score, and accuracy. Its ability to capture subtle linguistic nuances and generalize well across different data sources, as evidenced by its success on Reddit data, makes it a robust solution for this task.

Looking ahead, there are several exciting avenues for future exploration. Fine-tuning BERT with a larger, domain-specific dataset could further enhance its accuracy and sensitivity to mental health-related language patterns. Integrating user-specific contextual information, such as historical tweets or demographic data, could enhance the model's ability to identify individuals at risk. Furthermore, incorporating sentiment analysis beyond binary classification could provide a more nuanced understanding of users' emotional states, allowing for more targeted interventions. The project could also extend to other social media platforms to ensure a comprehensive analysis of individuals' digital footprints. Lastly, collaborating with mental health professionals to validate the model's predictions and ensure ethical considerations are met is crucial, given the sensitivity of mental health-related data.

Reference

1.Saleem, S., Prasad, R., Vitaladevuni, S., Pacula, M., Crystal, M., Marx, B., Sloan, D., Vasterling, J. and Speroff, T., 2012, December. Automatic detection of psychological distress indicators and severity assessment from online forum posts. In *Proceedings of COLING 2012* (pp. 2375-2388).

2.Ramit Sawhney, Prachi Manchanda, Raj Singh, and Swati Aggarwal. 2018. A Computational Approach to Feature Extraction for Identification of Suicidal Ideation in Tweets. In *Proceedings of ACL 2018, Student Research Workshop*, pages 91–98, Melbourne, Australia. Association for Computational Linguistics.

3.M. Trotzek, S. Koitka and C. M. Friedrich, "Utilizing Neural Networks and Linguistic Metadata for Early Detection of Depression Indications in Text Sequences," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 588-601, 1 March 2020, doi: 10.1109/TKDE.2018.2885515.

4.S. Ghosh and T. Anwar, "Depression Intensity Estimation via Social Media: A Deep Learning Approach," in IEEE Transactions on Computational Social Systems, vol. 8, no. 6, pp. 1465-1474, Dec. 2021, doi: 10.1109/TCSS.2021.3084154.

5.Tadesse, Michael Mesfin, Hongfei Lin, Bo Xu, and Liang Yang. 2020. "Detection of Suicide Ideation in Social Media Forums Using Deep Learning" *Algorithms* 13, no. 1: 7. https://doi.org/10.3390/a13010007

6.Sekulić, Ivan, and Michael Strube. "Adapting deep learning methods for mental health prediction on social media." *arXiv preprint arXiv:2003.07634* (2020).

7. Jiang, Zheng Ping, Sarah Ita Levitan, Jonathan Zomick, and Julia Hirschberg. "Detection of mental health from reddit via deep contextualized representations." In *Proceedings of the 11th international workshop on health text mining and information analysis*, pp. 147-156. 2020.

8. F. Haque, R. U. Nur, S. A. Jahan, Z. Mahmud and F. M. Shah, "A Transformer Based Approach To Detect Suicidal Ideation Using Pre-Trained Language Models," 2020 23rd International Conference on Computer and Information Technology (ICCIT), DHAKA, Bangladesh, 2020, pp. 1-5, doi: 10.1109/ICCIT51783.2020.9392692.

9. Murarka, Ankit, Balaji Radhakrishnan, and Sushma Ravichandran. "Classification of mental illnesses on social media using RoBERTa." In *Proceedings of the 12th international workshop on health text mining and information analysis*, pp. 59-68. 2021.

10. Swc Wang. 2023. “depression-detection” GitHub. https://github.com/swcwang/depression-detection

11. Viritaromero. 2019. “Detecting-Depression-in-Tweets” Github. https://github.com/viritaromero/Detecting-Depression-in-Tweets

12. Aryan-r22. 2022. “Stress-Detection_Social-Media-Articles” github
https://github.com/aryan-r22/Stress-Detection_Social-Media-Articles

13. Saurabh shahane. ”Twitter-sentiment-dataset”. 2022, kaggle
https://www.kaggle.com/datasets/saurabhshahane/twitter-sentiment-dataset

14. Xu, Shuo, Yan Li, and Zheng Wang. "Bayesian multinomial Naïve Bayes classifier to text classification." In *Advanced Multimedia and Ubiquitous Engineering: MUE/FutureTech 2017 11*, pp. 347-352. Springer Singapore, 2017.

15. M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428.

16. Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9, no. 8 (1997): 1735-1780.

1. https://aclanthology.org/C12-1145.pdf
2. https://aclanthology.org/P18-3013/
3. https://ieeexplore.ieee.org/abstract/document/8580405
4. https://ieeexplore.ieee.org/abstract/document/9447025
5. https://www.mdpi.com/1999-4893/13/1/7
6. https://arxiv.org/abs/2003.07634
7. https://aclanthology.org/2020.louhi-1.16.pdf
8. https://ieeexplore.ieee.org/abstract/document/9392692
9. https://aclanthology.org/2021.louhi-1.7.pdf
10. https://github.com/swcwang/depression-detection
11. https://github.com/viritaromero/Detecting-Depression-in-Tweets
12. https://github.com/aryan-r22/Stress-Detection_Social-Media-Articles
13. https://www.kaggle.com/datasets/saurabhshahane/twitter-sentiment-dataset