TEXT ANALYSIS FOR MENTAL HEALTH

INTRODUCTION

Mental health issues have become increasingly prevalent, constituting a major source of distress in people's lives with a significant impact on societal health and well-being. Addressing these challenges necessitates innovative approaches, and in recent years, text analysis techniques have shown promising potential. This paper provides a comprehensive narrative review of text analysis applications in mental health over the past decade, focusing on diverse textual data sources such as social media posts, interviews, and clinical notes.
Using text analysis, a technique that entails the methodical investigation of written or spoken communication, to obtain insights into people's mental health status, is one important line of inquiry. The abundance of digital text data produced by social media, online platforms, and communication channels offers a wealth of resources for deciphering the subtleties of discourse surrounding mental health.

With the continuous progress of the Internet, there is an increasing demand for the development of intelligent and sophisticated systems capable of effectively addressing the identification of health-related issues on social media, such as the recognition of depression and anxiety. These systems primarily rely on machine learning techniques and need to proficiently handle the extraction of semantic and syntactic meaning from the texts shared by users on social media. The content generated by users on social media is characterised by its unstructured and unpredictable nature. While various machine learning-based systems on social media platforms have been recently introduced to identify health-related problems, the employed text representation and deep learning methods offer only limited insights into the diverse texts posted by users.

The analysis includes sentiment analysis, depression detection using naive and weighted approaches, and topic modelling using Latent Dirichlet Allocation (LDA). Machine learning models, specifically Support Vector Machines (SVM) and Naive Bayes, are implemented for sentiment classification. The paper evaluates the effectiveness of naive depression classification methods and discusses insights gained from the analysis.

BACKGROUND

The prevalence of mental health discussions on online platforms has grown substantially, emphasising the need for effective analysis and support mechanisms. Previous research has explored sentiment analysis in social media data, depression detection in text, and the application of machine learning for mental health-related tasks. However, the specific context of Dreaddit and the combination of sentiment analysis, depression detection, and topic modelling in a single study make this project unique.

RESEARCH QUESTIONS

- What are the prevalent sentiments expressed in mental health-related posts on Dreaddit?
- How effective are naive and weighted keyword analysis methods in detecting signs of depression?
- What are the underlying topics within mental health-related discussions on Dreaddit, revealed through topic modelling?

- How well do machine learning models perform in classifying sentiments expressed in these posts?

METHODOLOGY

❖ DATA COLLECTION

The dataset consists of posts from Dreaddit and is split into training and testing sets. This data was chosen due to its relevance to mental health discussions and the availability of labelled information, including sentiment labels and meanings.

The dataset consists of two main files, "dreaddit-train.csv" and "dreaddit-test.csv," which are loaded into a Pandas DataFrame named 'full.' The combined dataset is explored to understand its structure, including the number of rows and columns, categorical columns, and the presence of missing values.

	subreddit	post_id	sentence_range	text	id	label	confidence	social_timestamp	social_karma	syntax_ari	...	lex_dal_min_pleasantness
0	ptsd	8601tu	(15, 20)	He said he had not felt that way before, sugge...	33181	1	0.8	1.521614e+09	5	1.806818	...	1.000
1	assistance	8lbrx9	(0, 5)	Hey there r/assistance, Not sure if this is th...	2606	0	1.0	1.527010e+09	4	9.429737	...	1.125
2	ptsd	9ch1zh	(15, 20)	My mom then hit me with the newspaper and it s...	38816	1	0.8	1.535936e+09	2	7.769821	...	1.000
3	relationships	7rorpp	[5, 10]	until i met my new boyfriend, he is amazing, h...	239	1	0.6	1.516430e+09	0	2.667798	...	1.000
4	survivorsofabuse	9p2gbc	[0, 5]	October is Domestic Violence Awareness Month a...	1421	1	0.8	1.539809e+09	24	7.554238	...	1.000

5 rows × 116 columns

Let's breakdown the columns:

- subreddit: The subreddit to which the post belongs.
- post_id: Unique identifier for each post.
- sentence_range: The range of sentences in the post that are relevant to the dataset.
- text: The actual text content of the post.
- id: Another form of identifier for the post.
- label: A binary label (0 or 1) indicating a certain characteristic or classification.
- confidence: Confidence level associated with the label.
- social_timestamp: Timestamp related to the social aspect of the post.
- social_karma: The karma (upvotes - downvotes) associated with the post.
- syntax_ari: Automated Readability Index (ARI) related to the syntax of the
- text.lex_dal_min_pleasantness, lex_dal_min_activation, lex_dal_min_imagery: These columns likely represent minimum values related to pleasantness, activation, and imagery based on some lexical analysis. These values could be derived from sentiment analysis tools or linguistic databases.
- lex_dal_avg_activation, lex_dal_avg_imagery, lex_dal_avg_pleasantness: These columns likely represent average values related to activation, imagery, and

pleasantness based on lexical analysis. Again, these values could be obtained from sentiment analysis tools or linguistic databases.

- Social_upvote_ratio: This column represents the ratio of upvotes to total votes (upvotes + downvotes) on the social platform where the post is located. It gives an indication of the post's popularity or approval within the community.
- Social_num_comments: This column represents the number of comments on the post, indicating the level of engagement and discussion it has generated.
- Syntax_fk_grade: This column may represent the Flesch-Kincaid grade level, a measure of the readability of the text. It estimates the level of education a person needs to easily understand the text.
- Sentiment: This column appears to represent a sentiment score associated with the post, possibly ranging from negative to positive. The values in this column may be numerical, with negative values indicating negative sentiment and positive values indicating positive sentiment.

❖ DATA PREPROCESSING

Textual data undergoes thorough preprocessing to remove noise and standardise formats. This involves converting text to lowercase, removing URLs and HTML tags, eliminating digits and non-word characters, tokenization, removing stopwords, and stemming.

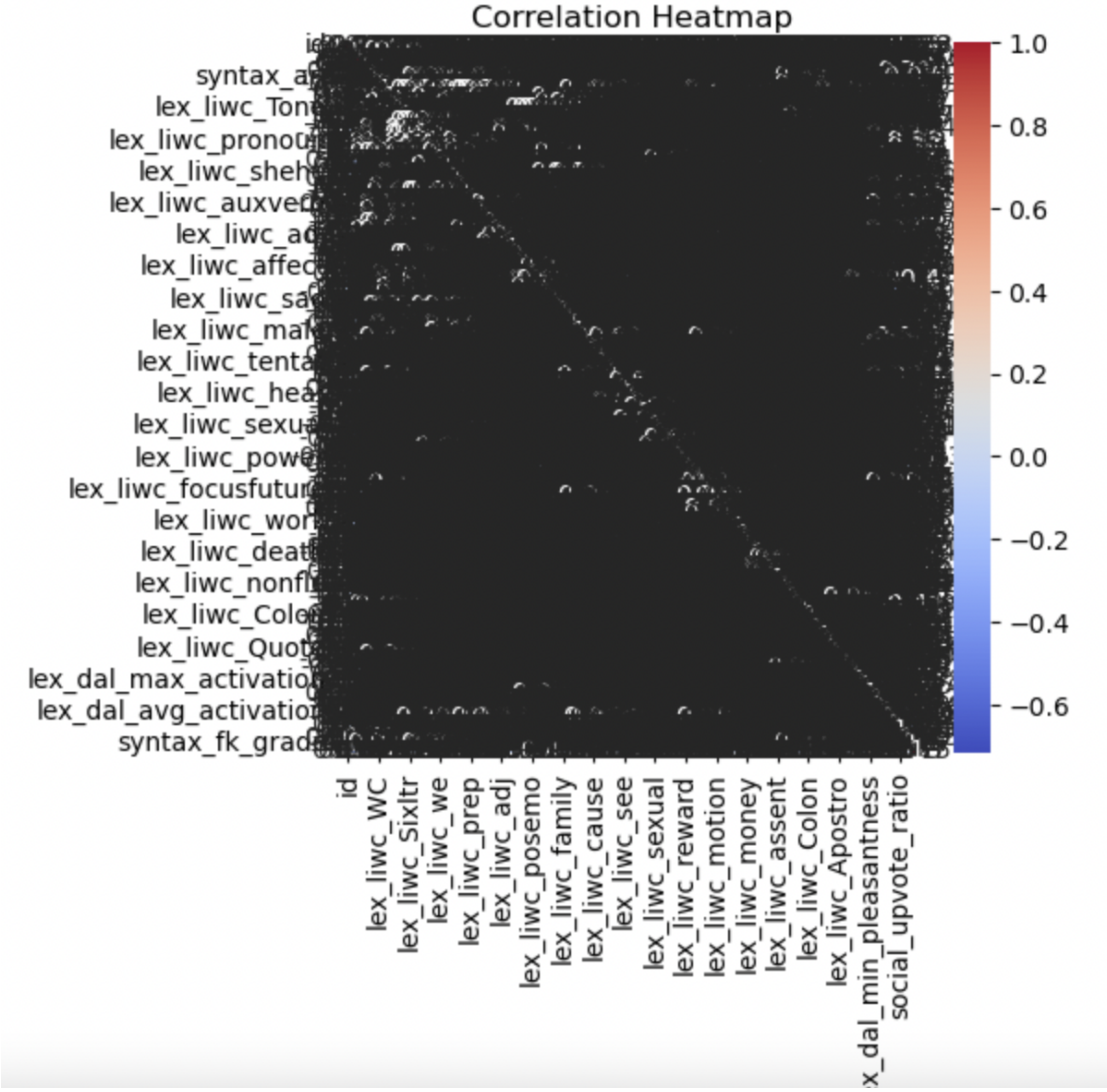
```
Rows : 3553
Columns : 116

Categorical columns :
subreddit           10
post_id           2929
sentence_range     194
text              3532
dtype: int64

Description :
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3553 entries, 0 to 714
Columns: 116 entries, subreddit to sentiment
dtypes: float64(107), int64(5), object(4)
memory usage: 3.2+ MB
None

Nulls:
subreddit                  0
post_id                    0
sentence_range             0
text                       0
id                         0
                          ..
lex_dal_avg_pleasantness   0
social_upvote_ratio        0
social_num_comments        0
syntax_fk_grade            0
sentiment                  0
Length: 116, dtype: int64
```

- **EXPLORATORY DATA ANALYSIS**

The EDA phase involves understanding the structure and characteristics of the dataset. This includes generating visualisations like correlation heatmaps, sentiment distributions, and word clouds to extract meaningful insights.

CORRELATION HEATMAP

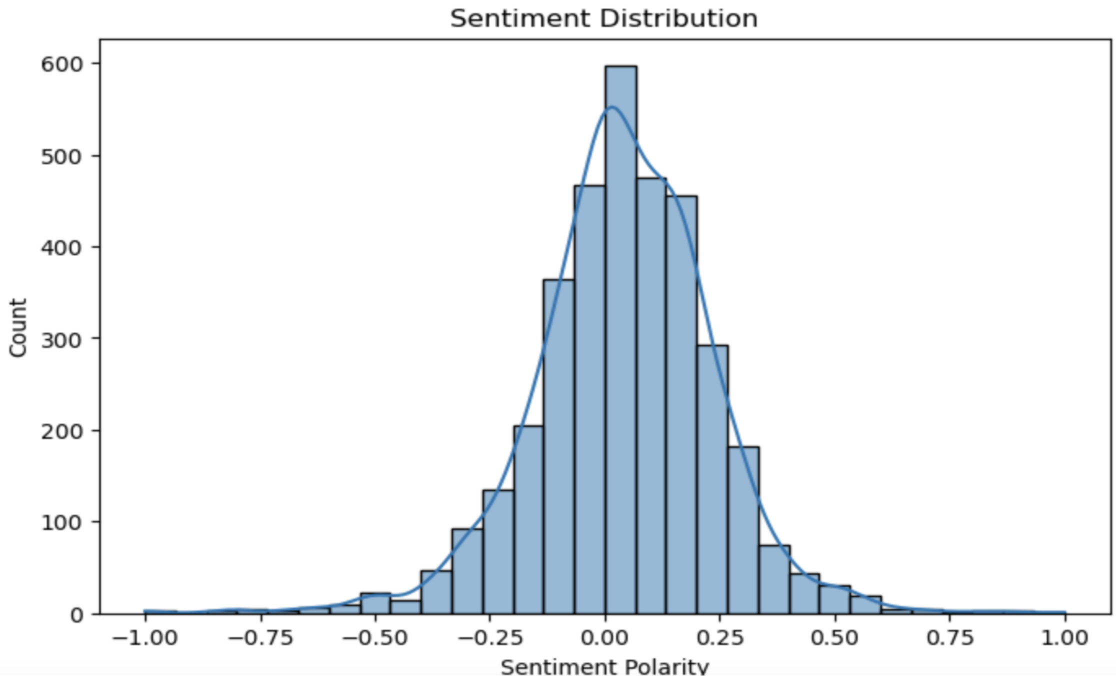
A correlation heatmap is generated to visualise the relationships between numerical features in the dataset.



SENTIMENT DISTRIBUTION

The distribution of sentiment polarity in the dataset is visualised using a histogram.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3553 entries, 0 to 714
Columns: 116 entries, subreddit to sentiment
dtypes: float64(107), int64(5), object(4)
memory usage: 3.2+ MB
None
```



The figure appears to be a histogram overlaid with a line plot, typically referred to as a density plot. It's used to visualise the distribution of sentiment polarity values in a dataset. Here's a breakdown of what this plot generally represents:

1. Sentiment Polarity: The values on the x-axis represent sentiment polarity, which is a measure of the emotional learning of text. This can range from -1.0 to 1.0, where -1.0 is extremely negative, 0 is neutral, and 1.0 is extremely positive.

2. Count: The y-axis represents the count of instances within the dataset that fall within each bin of sentiment polarity.

3. Histogram Bars: Each bar represents the frequency of sentiment polarity within a particular range (or bin). The height of the bar indicates how many records fall into that range.

4. Density Curve: The line plot overlaid on the histogram is likely a kernel density estimate (KDE), which is a smoothed version of the histogram and gives a sense of the probability density of the sentiment polarity.

5. Distribution Shape: The shape of the histogram is roughly bell-shaped (normal distribution), with a peak around just above 0, suggesting that most of the text is slightly positive in sentiment.

6. Skewness: There is a slight skew to the right (positive sentiment), which means there are more instances of positive sentiments than negative ones in the dataset.

7. Summary: The plot indicates that while there are occurrences of both negative and positive sentiments in the dataset, the majority of the text falls into the slightly positive range, with fewer instances of extreme sentiments on either the positive or negative side.

From the technical perspective of the output displayed above the plot:

- It indicates that the data is stored in a pandas DataFrame with 3553 entries (rows) and 116 columns.
- The DataFrame contains different types of data (`float64`, `int64`, `object`), which likely represent different features, including the sentiment scores.
- The memory usage of the DataFrame is slightly over 3 MB, which provides an indication of the dataset's size in memory.

SENTIMENT ANALYSIS

Sentiment analysis is performed using the TextBlob library. The sentiment polarity of each post is calculated and added as a new column in the DataFrame.

```
                                               text  sentiment
0  He said he had not felt that way before, sugge...  -0.002742
1  Hey there r/assistance, Not sure if this is th...   0.292857
2  My mom then hit me with the newspaper and it s...   0.011894
3  until i met my new boyfriend, he is amazing, h...   0.141671
4  October is Domestic Violence Awareness Month a...  -0.204167
```

SENTIMENT SUMMARY

Summary statistics and counts of sentiment polarity are presented to provide insights into the overall sentiment distribution.

```
Value Exploration:
Minimum Sentiment: -1.0
Maximum Sentiment: 1.0

Counts:
 0.000000    84
 0.250000    24
 0.100000    23
 0.150000    17
 0.125000    16
             ..
 0.200926     1
-0.009375     1
-0.045833     1
 0.244643     1
 0.136364     1
Name: sentiment, Length: 2627, dtype: int64
```

TEXT PREPROCESSING

Text data is preprocessed to remove noise and standardise the format. Steps include converting to lowercase, removing URLs and HTML tags, eliminating digits and non-word characters, tokenization, removing stopwords, and stemming.

```
0      said felt way before, sugget go rest ..trigger...
1      hey r/assistance, sure right place post this.....
2      mom hit newspap shock would this, know like pl...
3      met new boyfriend, amazing, kind, sweet, good ...
4      octob domest violenc awar month domest violenc...
                                     ...
710    horribl vivid nightmar everi night. sometim th...
711    also can't think without get angri jealou agai...
712    furthermore, told got realli seriou anxieti de...
713    here' link amazon wish list two item are: link...
714    keep us protected? alreadi told unwelcom perso...
Name: text, Length: 3553, dtype: object
```

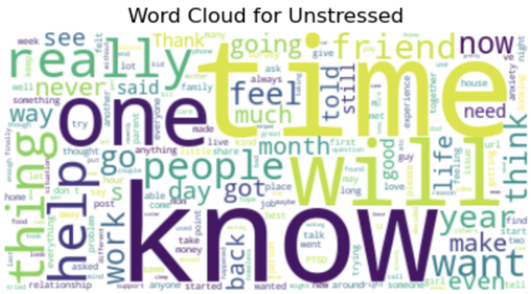
WORD CLOUDS

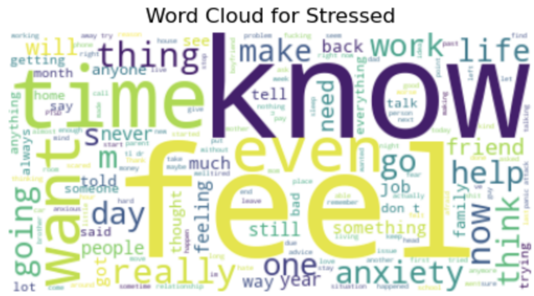
Word clouds are generated to visually represent the most frequent words in the dataset.

WORD FREQUENCY ANALYSIS

Word frequencies and word clouds for "Unstressed" and "Stressed" categories are explored.

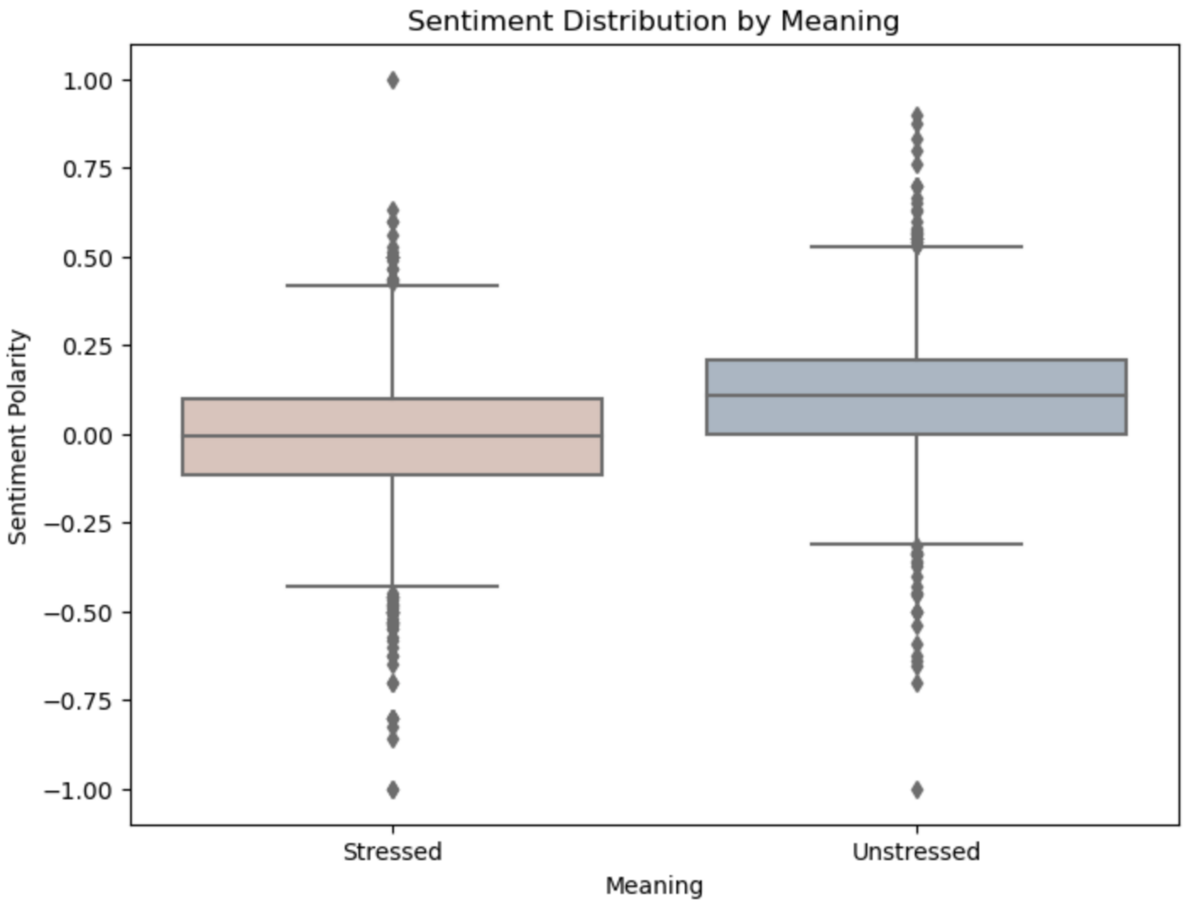
```
I          13232
to         10249
and         9579
the         7205
a           6461
my          4966
of          4479
in          3393
that        3345
for         2996
me          2926
was         2793
is          2601
have        2547
it          2509
with        2472
but         2274
this        1906
on          1756
he          1734
Name: text, dtype: int64
```





SENTIMENT DISTRIBUTION BY MEANING

A box plot is created to visualise the distribution of sentiment polarity across different meaning categories.



The figure appears to be a box plot, which is a standardised way of displaying the distribution of data based on a five-number summary: minimum, first quartile (Q1), median, third quartile (Q3), and maximum. It can also include outliers.

This particular box plot shows the sentiment polarity distribution across two different categories, labelled as "Stressed" and "Unstressed." Here's what the plot indicates about each category:

1. Stressed:
    - The sentiment polarity values for the "Stressed" category range from -1.00 to 1.00, with a median close to 0.
    - The box, representing the interquartile range (IQR), spans from approximately -0.25 to 0.25. This indicates that the middle 50% of the sentiment polarity values for the "Stressed" category are within this range.
    - There are outliers indicated by diamond shapes, which are sentiment polarity values that lie beyond the whiskers, or the typical range of the data.

2. Unstressed:
    - The sentiment polarity values for the "Unstressed" category also range from -1.00 to 1.00, but the median appears to be slightly higher than the "Stressed" category, indicating a somewhat more positive sentiment on average.
    - The IQR is narrower than in the "Stressed" category, suggesting less variability in sentiment among the "Unstressed" category.

- There are also outliers in this category, both on the lower and upper ends of the polarity spectrum.

General Observations:

- The distribution of sentiment polarity in both categories covers the full spectrum from -1.00 to 1.00, indicating a full range of sentiments from extremely negative to extremely positive.
- The "Stressed" category seems to have a wider spread of sentiment (wider IQR), indicating more variability in how strongly sentiments are expressed, compared to the "Unstressed" category.
- The presence of outliers in both categories suggests that there are a few instances with sentiments that are unusually strong, either positively or negatively, compared to the bulk of the data.

Overall, the box plot provides a visual comparison of sentiment distributions between two groups, allowing for insights into how sentiments differ depending on whether they are categorised as "Stressed" or "Unstressed."

DEPRESSION ANALYSIS

Two functions, naive_depression_analysis and weighted_depression_analysis, are implemented to perform depression analysis based on the presence and weighted sum of depressive keywords, respectively.

TOPIC MODELLING (LATENT DIRICHLET ALLOCATION)

Topic Modeling is a technique used in natural language processing and machine learning to identify topics present in a text corpus. One of the popular methods for topic modelling is Latent Dirichlet Allocation (LDA). Topic modelling using LDA is employed to uncover latent topics within the text data. This reveals the underlying themes and discussions prevalent in mental health-related posts on Dreaddit.

LDA assumes that there are K topics in the entire corpus, and each document is a mix of these topics.Each word in a document is attributable to one of the document's topics.LDA uses probability distributions to model the likelihood of a word belonging to a particular topic and the likelihood of a document being associated with a particular topic.The goal is to learn the topic distribution for each document and the word distribution for each topic.

```
Topic #1: help, job, work, ve, money, time, need, pay, family, don
Topic #2: got, time, just, home, didn, night, went, told, like, dad
Topic #3: ve, anxiety, like, years, time, just, ptsd, feel, really, year
Topic #4: like, just, don, feel, know, really, want, think, things, told
Topic #5: just, like, know, don, people, want, time, feel, work, day
```

- Topic 1: This topic seems to be about employment and financial concerns, as indicated by words like "help," "job," "work," "money," "need," and "pay." The presence of words like "family" may suggest that these concerns are related to family responsibilities or support.

- Topic 2: The words here like "got," "time," "home," "night," "went," "told," and "dad" seem to imply personal narratives or stories, potentially relating to daily life and family interactions.
- Topic 3: This topic is clearly associated with mental health, given the presence of "anxiety," "ptsd," "feel," "years," and "really." It might reflect discussions around personal experiences with anxiety and trauma over time.
- Topic 4: It includes words like "feel," "know," "really," "want," "think," and "things," which suggests introspection or expression of personal opinions and desires.
- Topic 5: This topic contains words that are more general and could be applicable in various contexts, but the inclusion of "work" and "day" next to terms like "know," "people," and "want" might indicate conversations about daily social interactions and experiences.

Topic 3, the LDA model would suggest that the document has a high probability of being about mental health issues.

LDA does not label the topics; it only provides the distribution of words. It's up to the user to interpret the topics based on the words' meanings and their relevance to the context of the documents. The model's success in accurately capturing meaningful topics depends on various factors, including the quality of the data preprocessing, the choice of parameters for the LDA (such as the number of topics K), and the coherence and distinctiveness of the topics themselves.

MACHINE LEARNING MODELS

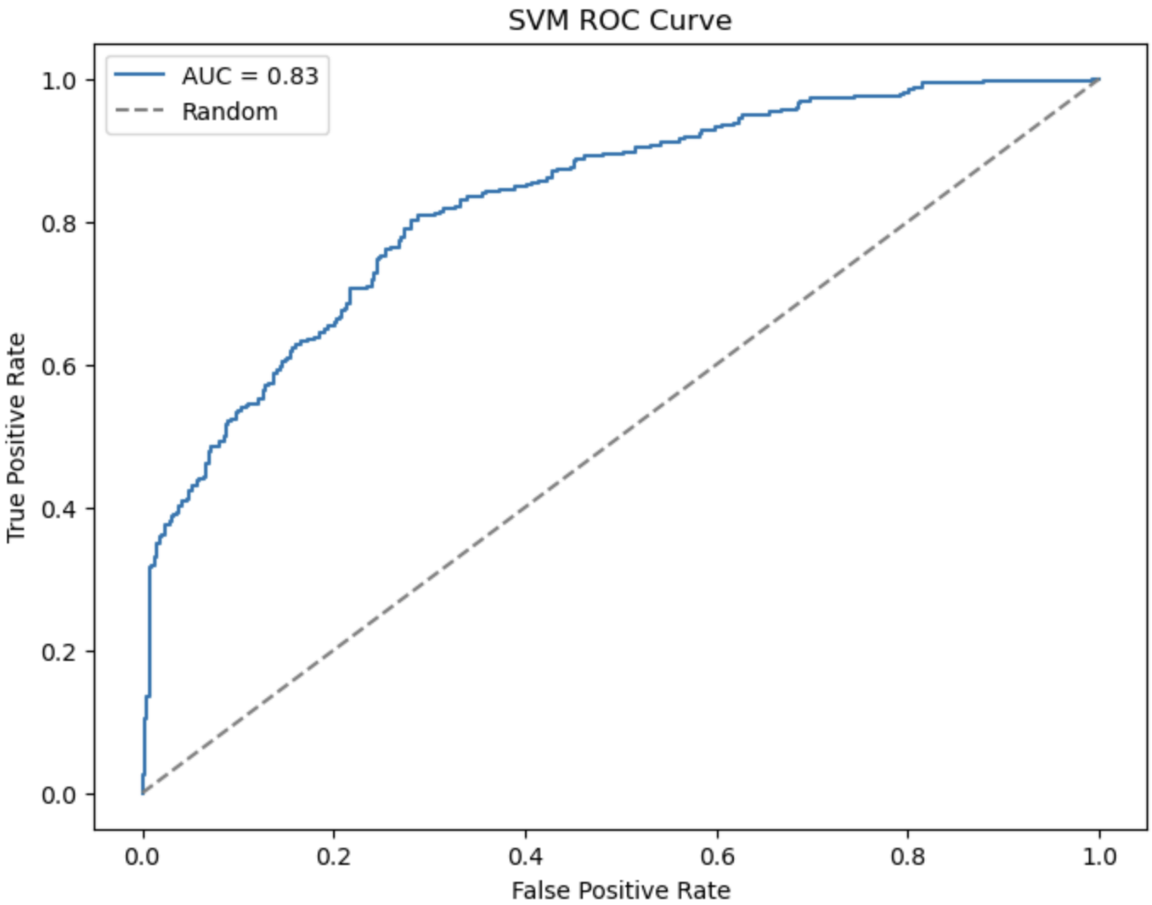
SVM and Naive Bayes models are implemented for sentiment classification. The TF-IDF vectorization technique is applied to convert text data into numerical features. GridSearchCV is used for hyperparameter tuning, and model performance is evaluated using accuracy, classification reports, and ROC curves.

- ❖ **SUPPORT VECTOR MACHINE (SVM)**

SVM, or Support Vector Machine, is a supervised machine learning model used for classification and regression tasks. However, it is most commonly used for classification problems. The main idea behind SVM is to find the best boundary (hyperplane) that separates the data into classes. This boundary is chosen to be the one where the margins, or the distances between the data points of different classes closest to the hyperplane, are maximised. In cases where data is not linearly separable, SVM can use kernel functions to transform the input space into a higher-dimensional space where a hyperplane can be used to separate the data.

```
SVM Model:
Accuracy: 0.749648382559775
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.74      0.76       388
           1       0.71      0.76      0.74       323

    accuracy                           0.75       711
   macro avg       0.75      0.75      0.75       711
weighted avg       0.75      0.75      0.75       711
```

- Accuracy: The model has an accuracy of about 74.96%, which means it correctly predicted the class for roughly 75% of the instances in the test dataset.
- Classification Report: This report includes several important metrics for evaluating the performance of a classification model for each class (labelled as 0 and 1, which are typically used for binary classification problems).
  - Precision: Precision is the ratio of true positive predictions to the total positive predictions made for a class. For class 0, the precision is 0.79, and for class 1, it is 0.71. This suggests the model is slightly better at correctly predicting instances of class 0 than class 1.
  - Recall: Recall, or sensitivity, is the ratio of true positive predictions to the actual number of positives. For class 0, the recall is 0.74, and for class 1, it is 0.76, indicating the model is slightly better at identifying all positive instances of class 1 than class 0.
  - F1-Score: The F1-score is the harmonic mean of precision and recall, a measure of the test's accuracy. For class 0, the F1-score is 0.76, and for class 1, it is 0.74. These scores show a balance between the precision and recall for each class.
  - Support: Support is the actual number of occurrences of the class in the dataset. There are 388 instances of class 0 and 323 instances of class 1 in the test data used to generate this report.
- Macro average and weighted average:
  - Macro Average: This averages the metric scores for each class without considering the support for each class. The macro average for precision, recall, and F1-score is 0.75, indicating average performance across both classes.
  - Weighted Average: This takes the support of each class into account when calculating the averages. In this output, the weighted average is the same as the macro average, which is somewhat unusual and indicates a balanced dataset or performance across the classes.



Receiver Operating Characteristic (ROC) curve, which is a graphical plot used to evaluate the performance of a binary classification system. The ROC curve is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings.

Here's what the different parts of the ROC curve indicate:

- True Positive Rate (TPR): Also known as recall or sensitivity, this is on the y-axis. It measures the proportion of actual positives that are correctly identified by the model. It's calculated as TPR = TP / (TP + FN), where TP is the number of true positives and FN is the number of false negatives.
- False Positive Rate (FPR): This is on the x-axis and measures the proportion of actual negatives that are incorrectly identified as positives by the model. It's calculated as FPR = FP / (FP + TN), where FP is the number of false positives and TN is the number of true negatives.
- AUC (Area Under the ROC Curve): This value is an aggregate measure of the model's performance across all classification thresholds. The AUC for this model is 0.83, which is quite good as it's close to 1. AUC values range from 0 to 1, where 1 indicates perfect classification and 0.5 indicates no discriminative power (equivalent to random guessing).
- Diagonal Dashed Line: This line represents the ROC curve of a random classifier (AUC = 0.5). The greater the distance of the ROC curve above this line, the better the model is at classification.

- ROC Curve: The blue line represents the ROC curve of the SVM model used for the text analysis. The curve shows the trade-off between sensitivity (TPR) and specificity (1-FPR) for different thresholds. A curve that hugs the top left corner indicates a good classification performance.

In the context of mental health text analysis, a higher AUC indicates that the SVM model is proficient in distinguishing between the two classes (which might be indicative of different mental health statuses, such as 'stressed' vs. 'unstressed', or 'positive' vs. 'negative' sentiment). An AUC of 0.83 suggests that the model has a high likelihood of correctly differentiating between texts associated with different mental health conditions or sentiments. This could be useful in automatically screening text to identify indications of mental health issues, which can then be used for further analysis by healthcare professionals or for providing resources to individuals who may benefit from them.

❖ NAIVE BAYES MODEL

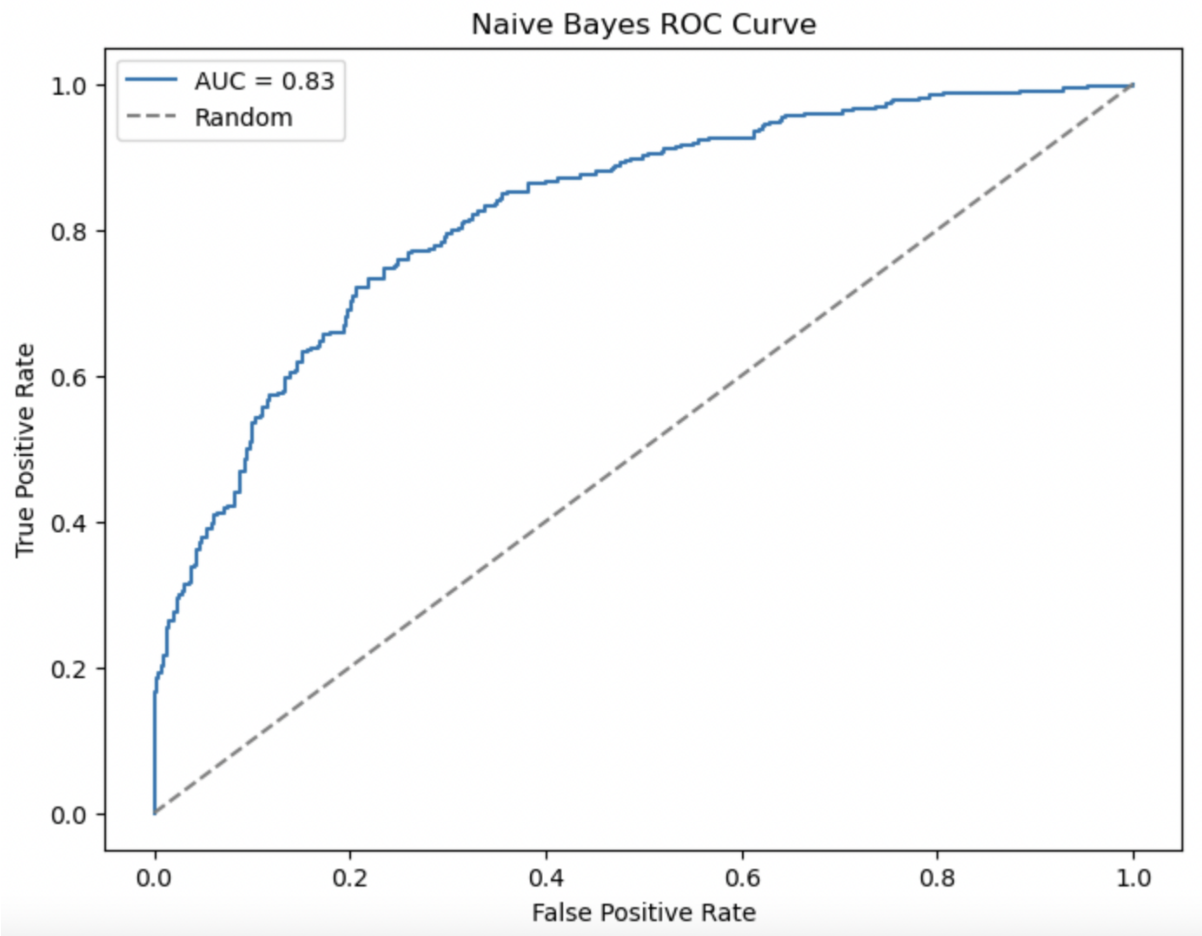
The Naive Bayes model is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It is particularly suited for high-dimensional datasets and is a popular method for text categorization tasks, where the features are related to word frequencies or presence.

```
Best Naive Bayes Model:
Accuracy: 0.7468354430379747
Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.84      0.78       388
           1       0.77      0.63      0.69       323

    accuracy                           0.75       711
   macro avg       0.75      0.74      0.74       711
weighted avg       0.75      0.75      0.74       711
```

- Accuracy: The model has an accuracy of approximately 74.68%, indicating that it correctly predicted the class for about three-quarters of the instances in the test dataset.
- Classification Report: This includes several key metrics:
- Precision: For class 0 (negative class), the precision is 0.73, meaning that when it predicts class 0, it is correct 73% of the time. For class 1 (positive class), the precision is 0.77, meaning it is correct 77% of the time when it predicts class 1.

- Recall: For class 0, the recall is 0.84, meaning that it correctly identifies 84% of all actual class 0 instances. For class 1, the recall is 0.63, meaning it correctly identifies 63% of all actual class 1 instances.
- F1-Score: This is a weighted average of precision and recall. For class 0, the F1-score is 0.78, and for class 1, it is 0.69. These scores are a measure of the model's accuracy and balance between precision and recall for each class.
- Support: This is the number of actual occurrences of each class in the specified dataset. Class 0 has 388 instances, and class 1 has 323.
- Macro Average: Averages the metric scores for each class without considering the support for each class. Here, both precision and recall have a macro average of 0.75, which gives an overall idea of the performance across both classes without considering class imbalance.
- Weighted Average: Takes the support of each class into account when calculating the averages. With a weighted average of 0.75 for precision, recall, and F1-score, the model shows consistent performance weighted by the number of instances in each class.

In summary, the Naive Bayes model seems to perform slightly better in identifying the negative class (class 0) than the positive class (class 1), as evidenced by the higher recall and F1-score for class 0. The overall accuracy and F1-score suggest that the model has a reasonable performance for this classification task. The balance between precision and recall, as reflected in the F1-score, is an important aspect of a model's performance, especially in applications where the cost of false positives and false negatives is different.



The curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold levels.

Here's a breakdown of the ROC curve and its features:

- True Positive Rate (TPR): Also known as recall or sensitivity, it is plotted on the y-axis. It measures the ratio of correctly predicted positive observations to all actual positives. It is calculated as TPR = TP / (TP + FN), where TP is the number of true positives and FN is the number of false negatives.
- False Positive Rate (FPR): Plotted on the x-axis, it measures the ratio of incorrectly predicted positive observations to all actual negatives. It is calculated as FPR = FP / (FP + TN), where FP is the number of false positives and TN is the number of true negatives.
- AUC (Area Under the ROC Curve): The AUC score is a single scalar value that summarises the overall performance of the classification model. It ranges from 0 to 1. An AUC of 0.83, as shown in the plot, indicates a good predictive ability. A model with perfect performance would have an AUC of 1.0, whereas a model with no predictive ability would have an AUC of 0.5, which is equivalent to random guessing.
- Diagonal Dashed Line: This represents the ROC curve of a classifier that makes random predictions. It serves as a baseline against which the classifier's performance is measured.

The ROC curve for the Naive Bayes model is significantly above the diagonal line, indicating that the model performs much better than random chance. Specifically, an AUC of 0.83 suggests that there is a high probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

The ROC curve is particularly useful for evaluating models in situations where there is an imbalance between the positive and negative classes. It provides a more nuanced picture of model performance compared to simple accuracy, as it shows how the true positive rate and false positive rate trade off at different thresholds. This is important for tuning the model according to the specific costs of false positives and false negatives in the application context.

- ❖ **DIFFERENCE BETWEEN TWO MODELS**

To determine which model performed better between the SVM and Naive Bayes based on the information provided, we need to compare their respective metrics:

1. Accuracy:
   - SVM Model: Approximately 74.96%
   - Naive Bayes Model: Approximately 74.68%

The SVM has a slightly higher accuracy.

2. Precision, Recall, and F1-Score:
   - For the SVM Model, the precision and F1-scores for class 0 and class 1 are quite balanced, both around 0.74-0.76.
   - For the Naive Bayes Model, the precision for class 0 is lower (0.73), and the recall for class 1 is significantly lower (0.63) compared to the SVM model. The F1-scores also reflect this, with a lower F1-score for class 1 (0.69) in the Naive Bayes Model.
3. AUC (Area Under the ROC Curve):
   - Both models have the same AUC of 0.83, indicating that their ability to distinguish between the classes is quite similar.

When comparing the performance of the two models, the SVM appears to have a slight edge in terms of balanced precision and recall across both classes, as indicated by the classification report. However, the Naive Bayes model is not far behind and has an equivalent AUC score.

In summary, while the SVM model has marginally better performance metrics across most categories, the choice of the model may also depend on the specific context in which it's being applied. For instance, if the cost of a false negative is high (e.g., in a medical diagnosis scenario where missing a true positive case can be very costly), then the model with the higher recall for the positive class would be preferred. If the application demands a faster model (e.g., for real-time predictions), Naive Bayes is known to be faster and could be chosen despite a slight drop in performance metrics.

Moreover, the AUC being the same for both models suggests that with proper threshold tuning, both models could potentially be adjusted to achieve similar performance for the specific needs of the task at hand. It's also important to consider other factors like model interpretability, training time, and scalability to the specific use case.

CONCLUSION

This project demonstrates a comprehensive analysis of mental health-related text data, covering sentiment analysis, depression detection, topic modelling, and machine learning models for sentiment classification. The insights gained from this analysis can be valuable for understanding the mental well-being of individuals expressing themselves on the Dreaddit platform. The implemented models can aid in automating the identification of sentiment and potential signs of distress, contributing to mental health monitoring and support.

FUTURE WORK

Future work may involve expanding the dataset to include more diverse sources, refining the depression detection models, and exploring advanced natural language processing techniques. Additionally, incorporating user interactions, temporal analysis, and further fine-tuning of machine learning models can enhance the overall understanding of mental health discourse on online platforms. The insights gained from this project can serve as a foundation for developing more sophisticated tools and interventions in the field of mental health analysis and support.