Project Report

On

Mental Health Mapper using
Machine Learning and Natural Language Processing

Submitted in fulfillment for the award of **Post Graduate Diploma in Artificial Intelligence (PG-DAI)** from
CDAC ACTS (Pune)

Guided By:

Mr. Prakash Sinha

Presented By:

Tulsi Mundada	PRN:240340128035
Shweta Anil Pawar	PRN:240340128032
Ruth Gaikwad	PRN:240340128025
Osheen Vasudevan	PRN:240340128017

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

Tulsi Mundada	PRN:240340128035
Shweta Anil Pawar	PRN:240340128032
Ruth Gaikwad	PRN:240340128025
Osheen Vasudevan	PRN:240340128017

Have successfully completed their project on

"Mental Health Mapper using Machine Learning and Natural Language Processing"

Under the guidance of

Mr. Prakash Sinha

Project Guide   Project Supervisor

ACKNOWLEDGEMENT

This project “**Mental Health Mapper using Machine Learning and Natural Language Processing**” was a great learning experience for us and we are submitting this work to Advanced Computing Training School (CDAC ACTS).

We all are very glad to mention the name of *Mr. Prakash Sinha sir* for his valuable guidance to work on this project. His guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

We are highly grateful to *Mrs. Risha P.R. mam* (Manager, ACTS training Centre), C- DAC, for her guidance and support whenever necessary while doing this course Post Graduate Diploma in *Artificial Intelligence(PG-DAI)* through C-DAC ACTS, Pune.

Our most heartfelt thanks go to *Mrs. Srujana bhamidi mam*(Course Coordinator, PG- *DAI*) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in C- DAC ACTS, Pune.

Sincerely,

Tulsi Mundada
Shweta Anil Pawar
Ruth Gaikwad
Osheen Vasudevan

ABSTRACT

This project focuses on developing a robust framework for analyzing and predicting individual mental states, aimed at improving mental health outcomes. The process begins with the creation of a comprehensive dataset that includes relevant textual data from individuals, potentially including clinical notes, patient reports, and other sources of written information related to mental health. The dataset undergoes rigorous preprocessing to clean and standardize the data, ensuring it is suitable for analysis.

Key features representing mental health indicators are then extracted using a combination of Bag of Words (BoW) and TF-IDF (Term Frequency-Inverse Document Frequency) techniques. These methods capture the most significant words and phrases, providing a nuanced understanding of the textual data's underlying themes.

Next, a Multinomial Naive Bayes (NB) model is trained on this processed data to predict future mental states based on historical patterns. This model is particularly well-suited for handling the categorical nature of the data and has been chosen for its effectiveness in text classification tasks.

For testing, speech-to-text data is integrated into the analysis. Patient data is examined individually, allowing for personalized insights and predictions. The results are further analyzed to produce a scoreboard that visualizes the relationship between predicted outcomes and actual data.

Finally, the project culminates in offering personalized recommendations aimed at enhancing individual well-being, grounded in the data-driven insights generated through the analysis. This comprehensive approach not only predicts mental health trends but also provides actionable feedback for mental health management.

TABLE OF CONTENTS

1.INTRODUCTION AND OVERVIEW OF PROJECT

Purpose

The purpose of this project is to create a framework for analyzing and predicting individual mental states using machine learning. By utilizing a comprehensive dataset, key features are extracted through Bag of Words and TF-IDF techniques to capture essential aspects of mental health. Sentiment analysis is applied to quantify emotions, and a Multinomial Naive Bayes algorithm is used to predict future mental states based on historical data. The model is tested with speech-to-text data from patients, enabling personalized predictions. The results are visualized, providing data-driven insights and recommendations for improving individual well-being. This project aims to enhance mental health monitoring and support.

Aim and Objective

Aim:

The aim of this project is to develop a machine learning framework for predicting individual mental states using text data from speech-to-text conversions. The goal is to provide accurate insights and recommendations to enhance mental well-being.

Objective:

The objectives of this project are to first acquire and preprocess a comprehensive mental health dataset to ensure its suitability for analysis. Key features are then extracted using Bag of Words and TF-IDF techniques to capture essential text elements related to mental health. Sentiment analysis is performed to quantify emotional states from the textual data. A Multinomial Naive Bayes model is trained on these features to predict future mental states based on historical patterns. The model's accuracy is validated using speech-to-text data from individual patients. Finally, the project involves visualizing the relationship between predictions and data and providing actionable recommendations to enhance individual well-being.

2.OVERALL PROJECT DESCRIPTION

2.1.Introduction:

This project focuses on developing a predictive framework for analyzing and forecasting individual mental states using advanced machine learning techniques. Mental health is a critical aspect of overall well-being, and the ability to accurately predict changes in mental states can provide valuable insights for early intervention and support. The project utilizes a comprehensive dataset that reflects various aspects of mental health, allowing for a detailed analysis of emotional and psychological patterns.

The initial steps involve preprocessing the data to ensure its quality and relevance. Key features are then extracted using Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) techniques, which capture the essential textual elements related to mental health. Sentiment analysis is applied to this textual data, enabling the quantification of emotional states.

The core of the project lies in training a machine learning model, specifically a Multinomial Naive Bayes (NB) algorithm, to predict future mental states based on historical data. This predictive capability is further tested using speech-to-text data, making the analysis more personalized and applicable to real-world scenarios.

Finally, the project includes comprehensive data visualization to illustrate the relationships between predictions and the underlying data, culminating in actionable recommendations for enhancing individual well-being. This framework not only contributes to the field of mental health analysis but also provides practical tools for mental health professionals to better monitor and support their patients.

2.2. System Architecture

The system architecture for this project is designed to efficiently analyze and predict individual mental states using a structured flow of data processing, feature extraction, model training, and visualization. Below is an overview of the key components in the system architecture:

1. **Data Collection Layer:**
    - **Data Sources:** The project begins by gathering data from various sources, including patient records, surveys, and speech-to-text transcriptions.
    - **Data Storage:** The collected data is stored in a centralized database, ensuring easy access and management. The database is designed to handle both structured and unstructured data related to mental health.
2. **Data Preprocessing Layer:**
    - **Data Cleaning:** Raw data is cleaned to remove noise, handle missing values, and standardize formats.
    - **Normalization:** Text data is normalized by converting it to lowercase, removing stop words, and performing stemming or lemmatization.
    - **Data Splitting:** The dataset is divided into training and testing sets to enable model evaluation.
3. **Feature Extraction Layer:**
    - **Bag of Words (BoW):** Textual features are extracted using the Bag of Words technique, which converts text into a matrix of word counts.
    - **TF-IDF:** The Term Frequency-Inverse Document Frequency technique is applied to weigh the importance of words in the context of the entire dataset, enhancing the feature extraction process.
4. **Model Training Layer:**
    - **Algorithm Selection:** The Multinomial Naive Bayes (NB) algorithm is selected for training due to its effectiveness in handling text data.

- **Training Process:** The extracted features are used to train the Multinomial NB model on the training dataset, enabling it to learn patterns related to mental states.

5. **Model Testing and Validation Layer:**

   - **Speech-to-Text Integration:** For real-time testing, speech data from patients is converted to text and fed into the model.
   - **Validation:** The model's predictions are validated against the test dataset to assess its accuracy and reliability.

6. **Data Visualization and Analysis Layer:**

   - **Visualization Tools:** Various data visualization techniques are employed to display the relationship between predictions and the underlying data, making the analysis more comprehensible.
   - **Scoreboard:** A scoreboard is generated to track the model's performance metrics and overall predictive accuracy.

7. **Recommendation Layer:**

   - **Actionable Insights:** Based on the analysis, the system provides recommendations aimed at improving the individual's mental well-being.
   - **Reporting:** A comprehensive report is generated, summarizing the findings and suggestions for mental health professionals.

8. **User Interface Layer:**

   - **GUI Development:** A user-friendly graphical user interface (GUI) is developed to facilitate interaction with the system, enabling users to input data and access recommendations easily.

2.3. System Requirements

2.3.1 Software Requirements

To effectively implement and run the mental state prediction system, the following software components are required:

1. **Operating System:**
    - Windows 10/11, macOS, or a Linux distribution (e.g., Ubuntu) compatible with the software tools used.
2. **Programming Language:**
    - Python 3.7 or later, due to its extensive libraries and support for machine learning and data processing.
3. **Libraries and Frameworks:**
    - **Pandas and NumPy:** For data manipulation and numerical operations.
    - **NLTK or SpaCy:** For natural language processing tasks.
    - **Scikit-learn:** For implementing the Multinomial Naive Bayes algorithm and other machine learning tasks.
    - **TensorFlow or PyTorch (optional):** If deep learning methods are explored.
    - **Matplotlib and Seaborn:** For data visualization.
    - **Flask or Django:** For developing the web-based user interface (if applicable).
4. **Database:**
    - MySQL, PostgreSQL, or MongoDB for storing and managing the collected data.
5. **Integrated Development Environment (IDE):**
    - PyCharm, Visual Studio Code, or Jupyter Notebook for coding and testing.
6. **Speech-to-Text API:**
    - Google Cloud Speech-to-Text, IBM Watson, or another speech recognition service.
7. **Version Control:**

- Git for managing code versions and collaboration.

2.3.2 Hardware Requirements

To ensure optimal performance for data processing, model training, and real-time predictions, the following hardware specifications are recommended:

1. **Processor:**
    - Intel Core i5 or i7 (8th generation or later) or AMD Ryzen 5 or 7 with a minimum clock speed of 2.5 GHz.
2. **Memory (RAM):**
    - At least 16 GB of RAM to handle large datasets and computational tasks efficiently.
3. **Storage:**
    - SSD with at least 256 GB of available space for storing datasets, model files, and software tools.
4. **Graphics Processing Unit (GPU):**
    - A dedicated GPU (e.g., NVIDIA GTX 1060 or later) with at least 4 GB of VRAM for accelerating model training, especially if deep learning methods are used.
5. **Network:**
    - Stable internet connection for accessing online APIs, cloud services, and collaborative tools.
6. **Peripherals:**
    - Standard keyboard and mouse.
    - Monitor with a resolution of at least 1080p for clear visualization and coding

3. Dataset Collection and Preparation

3.1. Data Sources

The success of this project heavily relies on the quality and relevance of the data used. The primary data sources include:

- **Mental Health Surveys:** Structured surveys and questionnaires targeting mental health, typically collected through online platforms or healthcare institutions.
- **Social Media Platforms:** Publicly available textual data from platforms like Twitter, Reddit, and mental health forums, where individuals often share their thoughts and emotions.
- **Clinical Records:** Anonymized patient records from mental health clinics or hospitals, which may include notes from therapy sessions, patient histories, and other relevant medical documentation.
- **Speech-to-Text Transcripts:** Audio recordings of patient conversations converted into text format using speech recognition tools to capture verbal expressions of mental states.

3.2. Data Collection Methodology

The data collection methodology involves systematic procedures to ensure that the gathered data is comprehensive, relevant, and representative:

1. **Ethical Considerations:** Prior to data collection, ethical approvals and consents are obtained, especially when dealing with sensitive mental health data. The data is anonymized to protect patient privacy.
2. **Survey Distribution:** Mental health surveys are distributed to targeted demographics through online platforms, focusing on various mental health aspects such as anxiety, depression, and stress levels.
3. **APIs and Web Scraping:** Data from social media platforms and forums are collected using APIs or web scraping tools like BeautifulSoup or Scrapy, ensuring compliance with the platforms' terms of service.
4. **Clinical Data Extraction:** Collaborations with healthcare institutions allow access to anonymized clinical records, where data is extracted in a format suitable for analysis.

5. **Speech-to-Text Conversion:** Audio recordings are processed through speech-to-text APIs (e.g., Google Cloud Speech-to-Text) to generate text data, which is then added to the dataset.

3.3. Data Preprocessing

Data preprocessing is a crucial step that involves preparing the raw data for analysis by cleaning it, handling missing values, and ensuring consistency.

3.3.1. Data Cleaning

Data cleaning involves removing or correcting any inaccuracies or inconsistencies in the dataset. The process includes:

- **Removing Duplicates:** Duplicate entries are identified and removed to prevent skewing the analysis.
- **Text Normalization:** Converting all text data to a consistent format by lowering case sensitivity, removing special characters, and expanding contractions (e.g., "don't" to "do not").
- **Stop Words Removal:** Common words that do not carry significant meaning (e.g., "and," "the," "is") are removed to enhance the focus on meaningful content.
- **Noise Reduction:** Irrelevant data, such as advertisements or off-topic content from social media, is filtered out to ensure only relevant text is **analyzed.**

3.3.2. Handling Missing Data

Handling missing data is critical to maintaining the dataset's integrity and ensuring accurate analysis. The following methods are applied:

- **Imputation:** Missing values are filled in using statistical techniques such as mean, median, or mode imputation, especially for numerical data.
- **Deletion:** In cases where data is missing from entire rows or columns, and the missing data cannot be reliably imputed, those entries may be removed from the dataset.
- **Forward/Backward Filling:** For time-series data, missing values are filled using the previous or next available data point.
- **Handling Null Values:** Textual data with missing labels or content are either labeled as "unknown" or excluded from the analysis if it affects the overall results.

4.Feature Extraction Techniques

Feature extraction is a critical step in transforming raw data into a format that machine learning algorithms can effectively use. In this project, two primary techniques were employed to extract features from textual data, which are crucial for analyzing and predicting mental states.

4.1 Bag of Words (BoW)

The Bag of Words (BoW) technique is a fundamental method for converting textual data into numerical representations. It involves the following steps:

- **Vocabulary Creation:** A list of all unique words (tokens) present in the dataset is compiled. Each word represents a feature.
- **Word Count Vectorization:** Each document (e.g., patient transcript) is converted into a vector where each element corresponds to the frequency of a specific word in that document. This vector ignores grammar and word order, focusing solely on word occurrence.
- **Simplicity and Efficiency:** BoW is simple to implement and works well for many basic text classification tasks. However, it does not account for the context or importance of words, treating all words equally.

BoW was used to generate a feature matrix from the speech-to-text data and patient records. This matrix serves as the input for the machine learning model, helping it to recognize patterns associated with different mental states.

4.2 TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF is an advanced technique that improves upon the Bag of Words approach by considering not just the frequency of words but also their importance within the dataset:

- **Term Frequency (TF):** This measures how frequently a word appears in a document relative to the total number of words in that document. It highlights commonly used words in each document.
- **Inverse Document Frequency (IDF):** This aspect of the technique reduces the weight of words that appear frequently across many

documents, as they are likely less informative (e.g., common words like "the" or "and"). It boosts the importance of words that are more unique to specific documents.

- **TF-IDF Calculation:** The final TF-IDF value for each word is calculated by multiplying its TF value by its IDF value. This results in a weighted vector where words unique to certain documents have higher significance.

TF-IDF was applied to the textual data to enhance the feature set by capturing the importance of terms related to mental health. This method allows the model to focus more on words that provide deeper insight into individual mental states, leading to more accurate predictions.

5. Model Selection and Training

5.1. Selection of Multinomial Naive Bayes (NB)

The Multinomial Naive Bayes algorithm was selected due to its effectiveness in handling text-based data, particularly in classification tasks like sentiment analysis. This algorithm assumes that features follow a multinomial distribution, which aligns well with the Bag of Words and TF-IDF features extracted from the dataset. Its simplicity and efficiency make it suitable for real-time mental state prediction, ensuring quick responses with reasonable accuracy.

5.2. Model Training Procedure

The model training involves splitting the preprocessed dataset into training and testing sets. The Multinomial Naive Bayes algorithm is trained on the training set, where it learns the relationship between the extracted text features (Bag of Words and TF-IDF) and the corresponding mental states. During training, the model estimates the probabilities of different mental states given the observed features, refining its parameters to improve predictive accuracy.

5.3. Hyperparameter Tuning

To enhance the model's performance, hyperparameter tuning is performed using techniques like grid search or randomized search. Key parameters such as the smoothing parameter (alpha) are adjusted to find the optimal values that minimize prediction error. This tuning process ensures that the model generalizes well to unseen data, balancing bias and variance.

5.4. Model Validation

Model validation is carried out using cross-validation techniques to assess the model's robustness. The trained model is evaluated on the testing set, where metrics such as accuracy, precision, recall, and F1-score are calculated. This step is crucial to ensure that the model not only fits the training data well but also performs effectively on new, unseen data, confirming its suitability for predicting mental states in real-world applications.

6. Testing and Evaluation

6.1. Integration of Speech-to-Text Data

In the testing phase, the first step is to integrate speech-to-text data into the model. Speech recordings from individual patients are transcribed into text using a speech recognition tool. This textual data is then preprocessed and transformed into feature vectors using the same Bag of Words and TF-IDF techniques employed during the training phase. The seamless integration of this data ensures that the model can process real-time inputs for mental state prediction.

6.2. Testing with Individual Patient Data

Once the speech-to-text data is prepared, the model is tested on individual patient data. This involves feeding the processed text into the trained Multinomial Naive Bayes model to predict the mental state of the patient. Each patient’s data is analyzed separately to evaluate the model's ability to accurately predict their mental state based on the features extracted from their speech.

6.3. Performance Metrics and Evaluation

The model's performance is evaluated using a set of standard metrics, such as accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of how well the model is performing, particularly in distinguishing between different mental states. Additionally, confusion matrices are generated to visualize the performance across different classes. The evaluation process also involves analyzing any misclassifications to identify potential areas for model improvement. This thorough evaluation ensures that the model is reliable and effective in predicting mental states, which is crucial for its application in real-world scenarios.

7. Sentiment Analysis and Data Analysis

7.1. Sentiment Analysis Methodology

The sentiment analysis process begins by applying a sentiment analysis algorithm to the preprocessed textual data. The goal is to categorize the text into different sentiment classes, such as Suicidal, Happy, or Neutral(Normal). This is typically done using lexicon-based approaches, machine learning models, or deep learning techniques. The sentiment scores are generated for each piece of text, reflecting the emotional tone of the content.

7.2. Analysis of Emotional States

Once the sentiment scores are obtained, they are used to analyze the emotional states of individuals. This involves aggregating sentiment scores over time to identify patterns and trends in an individual's emotional state. By analyzing these patterns, insights into the person's mental well-being can be drawn, such as identifying periods of high stress or depression. The analysis also involves cross-referencing sentiment data with other features extracted from the text to provide a comprehensive view of the individual's mental state.

7.3. Creation of the Scoreboard

A scoreboard is created to visually represent the sentiment analysis results and emotional state trends for each individual. The scoreboard displays key metrics, such as average sentiment score, frequency of negative sentiments, and emotional stability over time. This visual representation makes it easier to track changes in emotional states and provides a quick overview of an individual's mental health status.

7.4. Predictive Data Analysis

Predictive data analysis is conducted to forecast future mental states based on historical data. The sentiment scores and other features extracted from the text are used as inputs to the Multinomial Naive Bayes model or other predictive models. The model predicts the likelihood of future emotional states, allowing for proactive interventions or recommendations to enhance well-being. The predictive analysis is validated against actual outcomes to refine the model's accuracy and reliability.

```python
def predict_mood(text):
    vec = vector.transform([text])
    result = model.predict(vec)[0]
    return result

def save_to_csv(data, filename='patient_data.csv'):
    df = pd.DataFrame(data)
    df.to_csv(filename, mode='a', header=not os.path.exists(filename), index=False)

def collect_patient_data():
    data = []
    for i in range(7):
        print(f"Day {i + 1}:")

        # Speech-to-text conversion
        r = sr.Recognizer()
        with sr.Microphone() as source:
            print("Please say something:")
            audio = r.listen(source)
            try:
                text = r.recognize_google(audio, language="en-US")
                print("You said: " + text)
                preprocessed_text = preprocess_text(text)
                print("preprocessedd: " + preprocessed_text)
                predicted_mood = predict_mood(preprocessed_text)
                print("predicted: " + predicted_mood)


            except sr.UnknownValueError:
                print("Google Speech Recognition could not understand your audio")
                text = ""
            except sr.RequestError as e:
                print("Could not request results from Google Speech Recognition service; {0}".format(e))
                text = ""
```

8. Recommendation System

8.1. Personalized Recommendations for Mental Well-being

The recommendation system is designed to provide personalized suggestions for enhancing an individual's mental well-being based on the analysis of their emotional states and predictive data. This involves identifying specific areas where an individual may benefit from interventions, such as stress management techniques, therapeutic practices, or lifestyle adjustments. The recommendations are tailored to address the unique emotional patterns and needs of each individual, aiming to improve their overall mental health.

8.2. Implementation of Recommendations

Implementing the recommendations involves integrating them into a user-friendly interface or system where individuals can easily access and act upon the suggestions. This may include developing a mobile app or web platform where users receive daily or weekly recommendations based on their current mental state and historical data. The implementation process also involves providing resources, such as links to mental health services, self-help tools, or educational content, to support individuals in following through with the recommendations.

8.3. Evaluation of the Recommendation System

The effectiveness of the recommendation system is evaluated through user feedback and performance metrics. This includes assessing how well the recommendations align with the individuals' actual mental health improvements and collecting feedback on the usability and relevance of the suggestions. Metrics such as user satisfaction, engagement with the recommendations, and observed changes in mental well-being are used to gauge the success of the system. Based on this evaluation, adjustments and enhancements may be made to better meet the needs of users and improve the overall effectiveness of the recommendations.

```
def visualize_and_assess(filename='patient_data.csv'):
    # Read the CSV file
    df = pd.read_csv(filename)

    # Count the occurrences of each mood
    mood_counts = df['predicted_mood'].value_counts()

    # Plot the pie chart
    plt.figure(figsize=(4, 6))
    plt.pie(mood_counts, labels=mood_counts.index, autopct='%1.1f%%', startangle=140)
    plt.title('Mood Distribution')
    plt.show()

    # Determine the most common mood
    most_common_mood = mood_counts.idxmax()

    # Assess mental health and provide recommendations
    if most_common_mood == 'happy':
        health_status = 'The patient is fit.'
        recommendations = [
            'Continue engaging in activities that bring you joy.',
            'Maintain a healthy lifestyle with balanced nutrition and regular exercise.',
            'Consider setting new personal goals and challenges to stay motivated.'
        ]
    elif most_common_mood == 'Normal':
        health_status = 'The patient is in a normal mental state.'
        recommendations = [
            'Keep up with your current routines and practices.',
            'Stay connected with friends and family to maintain social support.',
            'Regularly check in with yourself to ensure you're managing stress effectively.'
        ]
    elif most_common_mood == 'depression':
        health_status = 'The patient is depressed.'
        recommendations = [
            'Consider seeking support from a mental health professional.',
```

10. Data Visualization

10.1. Visualization Techniques

Data visualization techniques are used to present complex information in an understandable and interpretable manner. For this project, various techniques are employed to display the results of mental state predictions and associated data. These techniques may include:

- **Bar Charts and Histograms:** To show the distribution of different mental health states or sentiment scores.
- **Line Graphs:** To track changes in mental states over time.
- **Scatter Plots:** To visualize relationships between different features or variables.
- **Heatmaps:** To display correlation matrices or intensity of emotional states across different dimensions.
- **Pie Charts:** To represent proportions of different sentiment categories or mental health conditions.

10.2. Representation of Prediction vs. Actual Data

The relationship between predicted and actual mental states is visualized to assess the accuracy and reliability of the model. This involves creating:

- **Comparison Charts:** Such as side-by-side bar charts or grouped line graphs to compare predicted mental states with actual data.
- **Confusion Matrices:** To illustrate the performance of the Multinomial Naive Bayes (NB) model in classifying mental states correctly.
- **Error Plots:** To highlight discrepancies between predicted and actual values, providing insights into model performance and areas for improvement.

10.3. Interpretation of Visual Data

Interpreting visual data involves analyzing and drawing insights from the visual representations to make informed conclusions. This step includes:

- **Identifying Trends and Patterns:** Observing any recurring patterns or trends in the data that reveal insights about mental state predictions.
- **Evaluating Model Performance:** Assessing how well the predictions align with the actual data and identifying any systematic errors or biases.

- **Providing Insights:** Offering actionable recommendations based on the visualized data, such as adjustments to the prediction model or targeted interventions for individuals based on their mental health trends.

```
# Create the GUI window
root = tk.Tk()
root.title("Mental Health Dataset Visualization")

# Create a figure and canvas for the plot
fig = Figure(figsize=(8, 6), dpi=100)
ax = fig.add_subplot(111)
canvas = FigureCanvasTkAgg(fig, master=root)
canvas.draw()
canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=1)

# Define functions for each visualization
def plot_status_correlations():
    ax.clear()
    weekday_status_matrix = df.pivot_table(index="text", columns="predicted_mood", aggfunc="size", fill_value=0)
    corr_matrix = weekday_status_matrix.corr()
    sns.heatmap(corr_matrix, ax=ax, annot=True, cmap='coolwarm',cbar=False)
    ax.set_title("Status Correlations")
    canvas.draw()

def plot_day_of_week_distribution():
    ax.clear()
    df['day'].value_counts().plot(kind='bar', ax=ax)
    ax.set_title("Day of Week Distribution")
    canvas.draw()

def plot_status_over_time():
    ax.clear()
    df.groupby('day')['predicted_mood'].value_counts().unstack().plot(kind='line', ax=ax)
    ax.set_title("Status Over Time")
    canvas.draw()

def plot_word_cloud():
    ax.clear()
    wordcloud = WordCloud().generate(' '.join(df['text']))
```

11. GUI Development

11.1. Design of the Graphical User Interface

The design of the graphical user interface (GUI) involves creating a user-friendly layout that facilitates interaction with the mental state prediction system. Key considerations include:

- **Layout and Navigation:** Designing a clear and intuitive layout that organizes key features and functionalities. This might include sections for data input, results display, and recommendations.
- **Visual Design:** Ensuring a visually appealing design that aligns with the project's purpose, including consistent color schemes, fonts, and icons that enhance usability.
- **Interactive Elements:** Incorporating buttons, sliders, and dropdown menus for users to interact with the system, such as inputting new data or selecting different analysis options.

11.2. User Interface Implementation

The implementation of the user interface (UI) involves developing the actual interface based on the design. This step includes:

- **Development Tools:** Using programming languages and frameworks (e.g., Python with Tkinter or JavaScript with React) to build the interface.
- **Integration with Backend:** Connecting the UI to the backend processes of the system, including data preprocessing, feature extraction, model predictions, and visualization components.
- **Testing:** Conducting user testing to ensure that the interface functions correctly, is free of bugs, and meets user expectations.

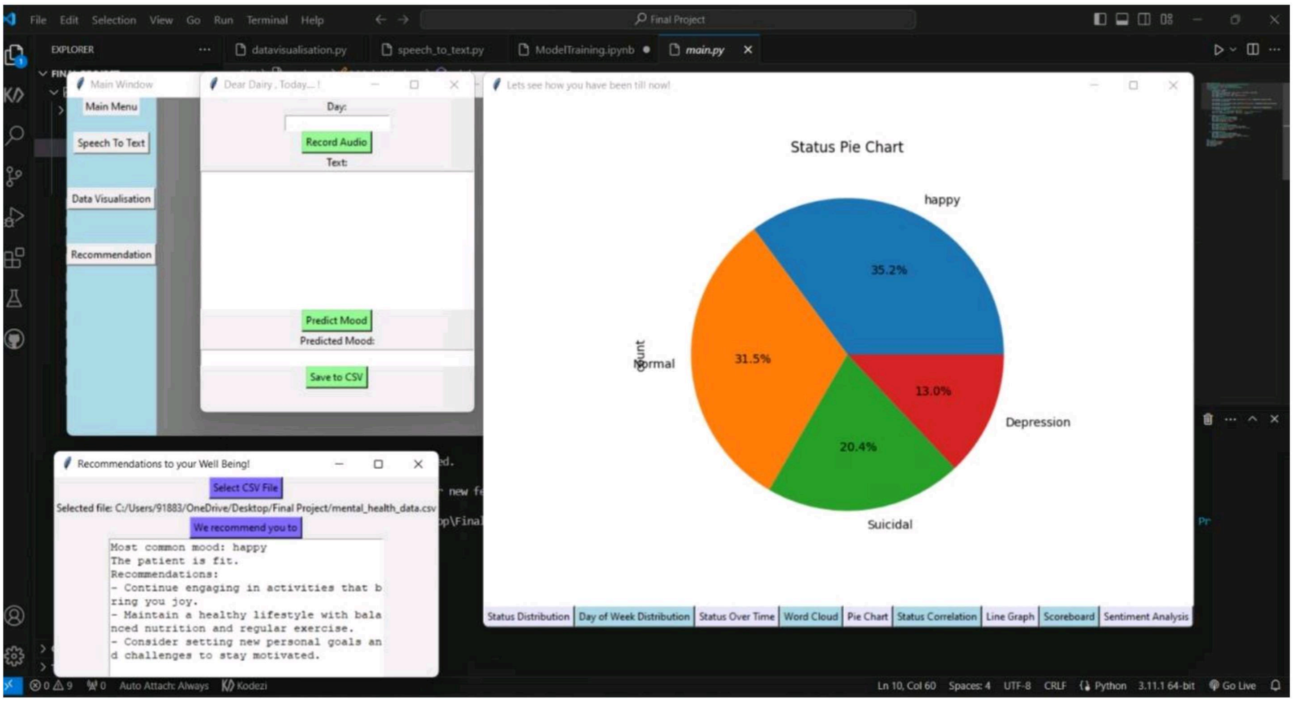
11.3. Enhancing User Experience

Enhancing user experience (UX) focuses on improving the usability and overall satisfaction of the GUI. This involves:

- **Feedback Mechanisms:** Implementing features like tooltips, help sections, and feedback forms to assist users and gather their input for improvements.

- **Performance Optimization:** Ensuring that the interface operates smoothly, with quick response times and minimal loading delays.
- **Accessibility:** Making the interface accessible to users with different needs, such as including support for screen readers and ensuring that all interactive elements are easily navigable.

These steps ensure that the GUI not only provides essential functionalities but also offers a seamless and engaging experience for users interacting with the mental state prediction system.



12. Project Implementation

12.1. Workflow of the Project

The workflow of the project is designed to systematically address the problem of predicting individual mental states using machine learning. The project begins with the collection and preparation of a comprehensive dataset. Following this, data preprocessing is carried out to clean and structure the data effectively. Key features are then extracted using the Bag of Words and TF-IDF techniques. Subsequently, a Multinomial Naive Bayes (NB) model is trained and tested. The testing phase includes converting speech to text, applying the trained model to individual patient data, and evaluating its performance. Data analysis and visualization are performed to interpret results and provide insights. Finally, recommendations are developed based on the analysis, and a graphical user interface (GUI) is created to present the results and recommendations to users.

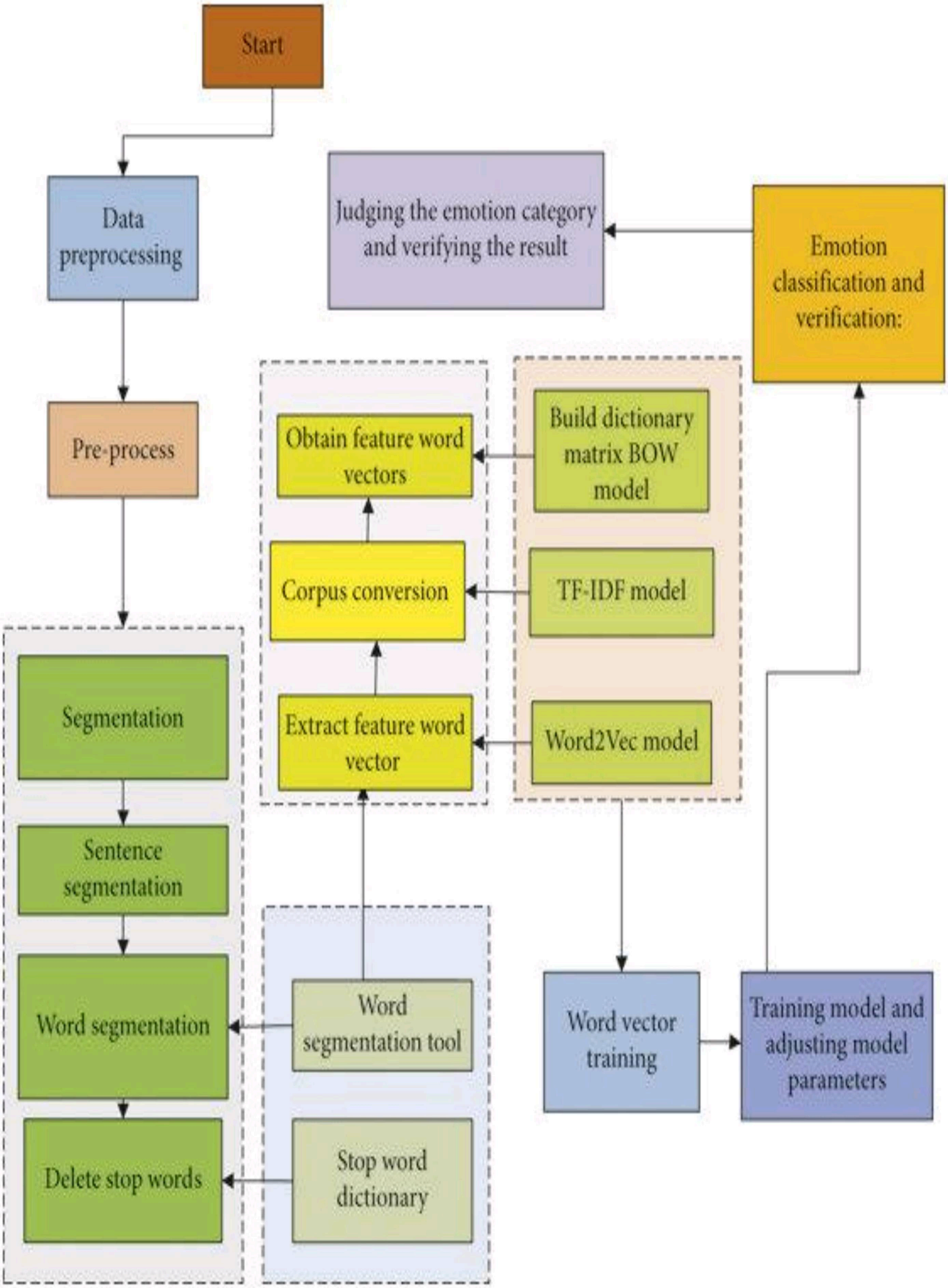
12.2. Step-by-Step Implementation

1. **Dataset Collection and Preparation**: Gather diverse and comprehensive datasets that reflect various mental health aspects. Prepare the data for analysis through cleaning and preprocessing to ensure accuracy and relevance.
2. **Data Preprocessing**: Clean the data by handling missing values and normalizing data. This step includes removing irrelevant information and standardizing formats to ensure consistency.
3. **Feature Extraction**: Use the Bag of Words and TF-IDF techniques to extract meaningful features from the text data. These features are critical for the machine learning model to understand and classify mental states.
4. **Model Training**: Train the Multinomial Naive Bayes (NB) model using the extracted features. This step involves fitting the model to historical data to learn patterns and relationships in mental state indicators.
5. **Testing with Speech-to-Text Data**: Convert speech data to text and test the trained model with this data. This step validates the model's ability to handle and analyze real-world, unstructured data from patients.
6. **Data Analysis and Visualization**: Analyze the results of the model predictions and use various visualization techniques to represent the relationship between predictions and actual data. This helps in understanding patterns and trends.

7. **Recommendation Development**: Based on the analysis, develop recommendations for improving individual mental well-being. These recommendations are derived from insights gained through data analysis.
8. **GUI Development**: Design and implement a graphical user interface (GUI) to make the system user-friendly. The GUI allows users to interact with the system, input data, view results, and receive recommendations.

12.3. Integration of All Components

The integration involves combining all individual components into a cohesive system. Ensure seamless data flow from collection through preprocessing to feature extraction and model training. Integrate the machine learning model with the GUI to allow for real-time predictions and analysis. Ensure that the system components work together efficiently and accurately, providing a unified user experience. Final testing is conducted to verify that the system meets all functional requirements and performs reliably across various scenarios.


Start
Data preprocessing
Pre-process
Segmentation
Sentence segmentation
Word segmentation
Delete stop words
Word segmentation tool
Stop word dictionary
Extract feature word vector
Corpus conversion
Obtain feature word vectors
Build dictionary matrix BOW model
TF-IDF model
Word2Vec model
Judging the emotion category and verifying the result
Emotion classification and verification:
Word vector training
Training model and adjusting model parameters

13. Conclusion and Future Work

13.1. Summary of Results and Findings

The project successfully developed a framework for analyzing and predicting individual mental states using a combination of machine learning and text analysis techniques. After preprocessing and feature extraction using Bag of Words and TF-IDF, sentiment analysis quantified emotional states. The Multinomial Naive Bayes (NB) model demonstrated its capability to predict future mental states based on historical patterns. Data visualization techniques effectively illustrated the relationship between predictions and actual data, providing actionable insights for individual well-being. The system's predictive accuracy and recommendations offer valuable support for mental health management.

13.2. Limitations of the Study

Despite the project's successes, several limitations were noted. The dataset may not cover all possible mental health conditions, potentially affecting the model's generalizability. The reliance on textual data from speech-to-text conversions may introduce errors or misinterpretations. Additionally, the model's performance may vary with different types of input data or in diverse demographic groups. The study also faced challenges in balancing the model's complexity and interpretability.

13.3. Suggestions for Future Research

Future research should focus on expanding the dataset to include a wider range of mental health conditions and diverse populations to improve the model's generalizability. Enhancing the speech-to-text accuracy and integrating additional data types, such as physiological or behavioral indicators, could provide a more comprehensive analysis of mental states. Investigating advanced machine learning techniques and algorithms may also yield improved predictive performance. Furthermore, developing and testing new methods for delivering personalized recommendations could enhance individual well-being support.

14. References

➢https://www.analyticsvidhya.com/blog/2022/06/mental-health-prediction-using-machine-learning/

➢https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8603338/

➢https://www.geeksforgeeks.org/what-is-sentiment-analysis/

➢https://www.cio.com/article/189218/what-is-sentiment-analysis-using-nlp-and-ml-to-extract-meaning.html