UNLOCKING THE MINDS: ANALYZING MENTAL HEALTH WITH NLP

Major Project documentation submitted to

# JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

**(DATA SCIENCE)**

Submitted By

**DIVYA SRI KUSURI (21UK1A6799)**

**LIKHITHA RAGAM (21UK1A6783)**

**SAIRAM SANKOJI (21UK1A6785)**

**PURNACHAND KONGALA (22UK5A6709)**

Under the guidance of

**K. SOUMYA**

Asst. Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# (DATA SCIENCE)

## VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(DATA SCIENCE)**

# VAAGDEVI ENGINEERING COLLEGE WARANGAL



**CERTIFICATE**

This is to certify that the Major project report entitled “UNLOCKING THE MINDS: ANALYZING MENTAL HEALTH WITHNLP” is being submitted by **DIVYA SRI KUSURI(21UK1A6799),LIKHITHA RAGAM(21UK1A6783),SAIRAM SANKOJI(21UK1A6785),PURNACHAND KONGALA(21UK5A6702)**,partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024-2025.

**Project Guide**    **Head of the Department**

**K. SOUMYA Dr. K. SHARMILAREDDY**

**(Asst. professor) (Professor)**

# ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. Prasad Rao**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this major project in the institute.

We extend our heartfelt thanks to **Dr. K. SHARMILAREDDY,** Head of the Department of CSD, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the mini project.

We express heartfelt thanks to the guide **K. SOUMYA**, Asst. Professor, Department of CSD for her constant support and giving necessary guidance for completion of this major project.

Finally, we express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experiencing throughout thesis.

**DIVYA SRI KUSURI (21UK1A6799)**

**LIKHITHA RAGAM (21UK1A6783)**

**SAIRAM SANKOJI (21UK1A6785)**

**PURNACHAND KONGALA (22UK5A6709)**

**Unlocking the Minds: Analysing Mental Health with NLP**

Mental health is a critical aspect of overall well-being, yet it remains a complex and often stigmatized issue. With advancements in Natural Language Processing (NLP), there is a promising opportunity to gain deeper insights into mental health conditions through the analysis of text data. This project aims to utilize NLP techniques to unlock valuable insights from various sources of textual data related to mental health

Leveraging NLP techniques to analyse textual data related to mental health holds immense potential for gaining insights into the complexities of mental health conditions. By unlocking the linguistic patterns and sentiments expressed in textual data, this project aims to contribute to the advancement of mental health research, support, and intervention efforts.

Scenario 1:

Project Description: "Unlocking the Minds: Analysing Mental Health with NLP" harnesses Natural Language Processing (NLP) techniques to extract insights from textual data pertaining to mental health. Through the analysis of linguistic patterns and sentiments expressed in various sources of text, this project seeks to deepen understanding of mental health conditions, contributing to research, support, and intervention efforts. Scenarios: Social Media Sentiment Analysis: Researchers use NLP algorithms to analyse social media posts related to mental health. By examining the language and sentiments expressed in these posts, they identify trends, common struggles, and support networks within online communities. This insight can inform mental health advocacy campaigns, help target interventions, and provide valuable resources for individuals seeking support in online spaces.

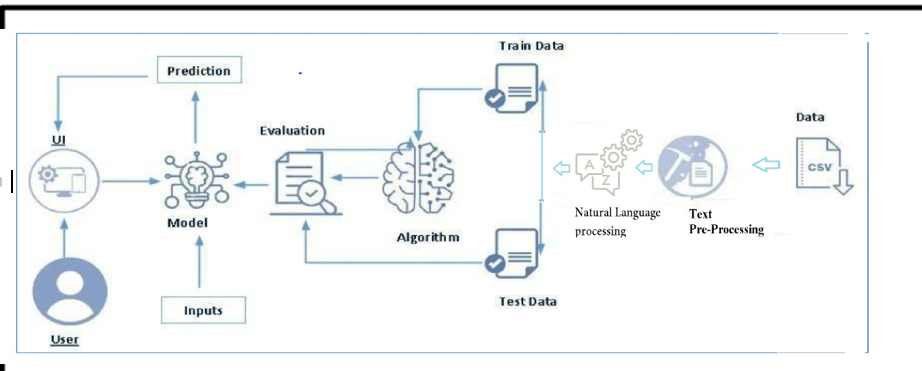
Scenario 2:

Therapist-Patient Communication Analysis: In a clinical setting, therapists utilize NLP tools to analyse transcripts of therapy sessions. By examining language patterns, emotions, and topics discussed, therapists gain deeper insights into patients' mental health struggles, progress, and treatment needs. This analysis can inform personalized treatment plans, identify areas for therapeutic focus, and enhance the effectiveness of therapeutic interventions.

Scenario 3:

Academic Research Paper Analysis: Researchers in the field of psychology and psychiatry employ NLP techniques to analyse academic research papers on mental health topics. By extracting key concepts, trends, and findings from a large corpus of scholarly literature, researchers gain a comprehensive understanding of current knowledge gaps, emerging areas of research, and effective treatment approaches. This analysis can guide future research directions, inform evidence-based practice, and contribute to the advancement of mental health science.

**Technical Architecture**



**Project Flow**

* User interacts with the UI to enter the input.
* Entered input is analyzed by the model which is integrated.
* Once model analyses the input the prediction is showcased on the UI To accomplish this, we have to complete all the activities listed below,
* Define Problem / Problem Understanding or Specify the business problem
* Business requirements
* Literature Survey
* Social or Business Impact.
* Data Collection & Preparation
* Collect the dataset
* Data Preparation
* Exploratory Data Analysis
* Descriptive statistical
* Visual Analysis
* Model Building
* Training the model in multiple algorithms
* Testing the model
* Performance Testing
* Testing model with multiple evaluation metrics
* Model Deployment
* Save the best model

Integrate with Web Framework

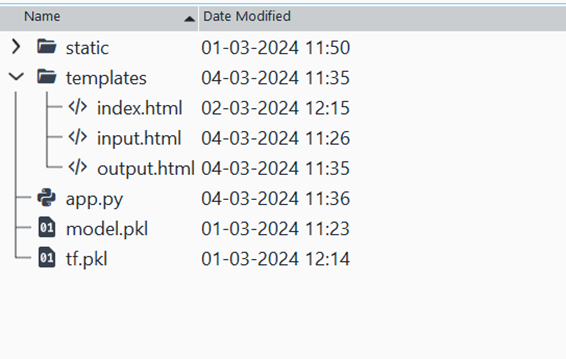
**Prior Knowledge**

You must have prior knowledge of following topics to complete this project.

* ML Concepts
* Supervised learning: <https://www.javatpoint.com/supervised-machine-learning>
* Unsupervised learning: <https://www.javatpoint.com/unsupervised-machine-learning>
* Decision tree: [https://www.javatpoint.com/machine-learning-decision-tree-classification](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm)- [algorithm](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm)
* Random forest: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>
* Xgboost:[https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-](https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/) [understand-the-math-behind-xgboost/](https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/)
* Evaluation metrics: [https://www.analyticsvidhya.com/blog/2019/08/11-important-model-](https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/) [evaluation-error-metrics/](https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/)
* NLP:-https://[www.javatpoint.com/nlp](http://www.javatpoint.com/nlp)
* Flask Basics: <https://www.youtube.com/watch?v=lj4I_CvBnt0>

**Technical Architecture**

Create the Project folder which contains files as shown below



* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
* model.pkl is our saved model. Further we will use this model for flask integration.
* tf.pkl is our TFIDF converter file.

**Data Collection & Preparation**

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

**Collect the dataset**

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

Link: <https://www.kaggle.com/code/swathiunnikrishnan/mental-health-analysis-using-nlp/input>

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analyzing techniques.

Note: There are a number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

Mental Health Analysis using NLP | Kaggle..

Explore and run machine learning code with Kaggle Notebooks | Using data from Mental Health Corpus..

<https://www.kaggle.com/code/swathiunnikrishnan/mental-health-analysis-using-nlp/input>

**Importing the libraries**

Import the necessary libraries as shown in the image.

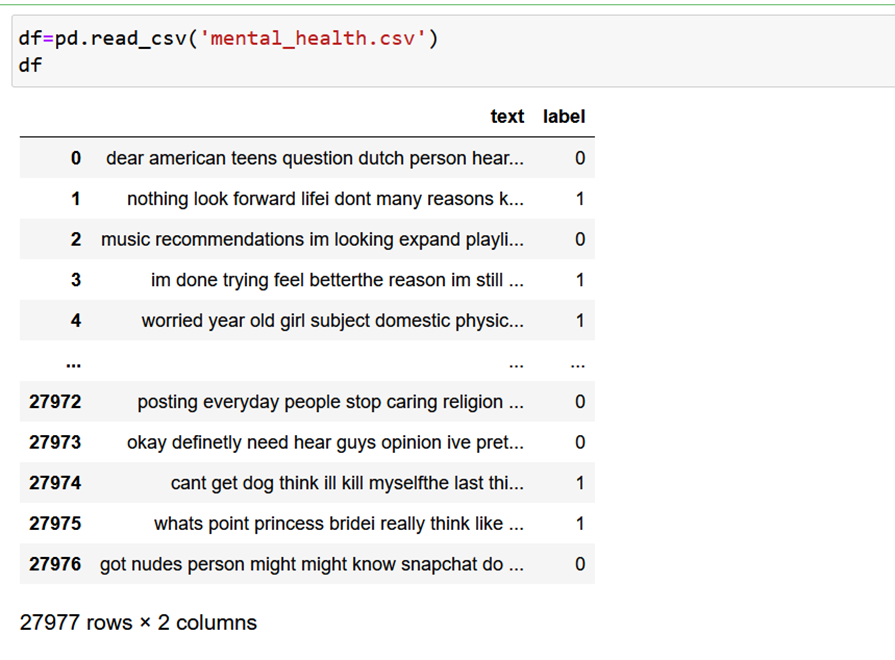


**Read the Dataset**

Activity 3: Read the Dataset

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of the csv file.



**Data Preparation**

As we have understood how the data is, let's pre-process the collected data.

The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

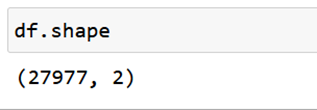
* Handling missing values
* Handling Duplicate Values
* Handling Imbalance Data

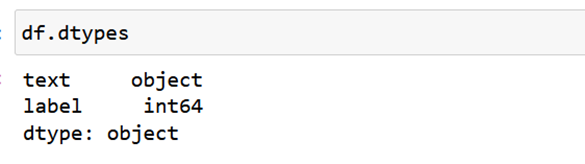
Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

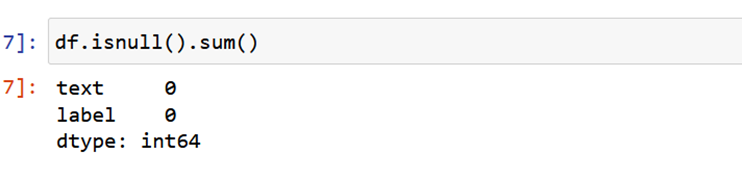
**Handling missing values**

Let’s find the shape of our dataset first. To find the shape of our data, the df.shape method is used. To find the data type, df.dtypes function is used

For checking the null values, df.isnull() function is used. To sum those null values we use df.isnull().sum() function. From the below image we found that there are no null values present in our dataset. So we can skip handling the missing values step.







**Handling Duplicate Values**

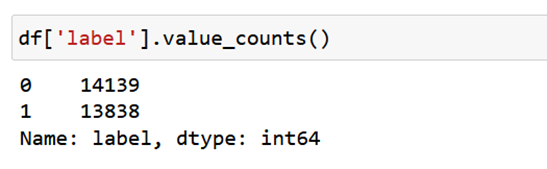
Handling duplicate data is a common challenge in data processing Once duplicates are identified, deduplication techniques can be applied to remove redundant entries.

To drop  the duplicate values we use drop\_duplicated().



**Handling Imbalance Data**

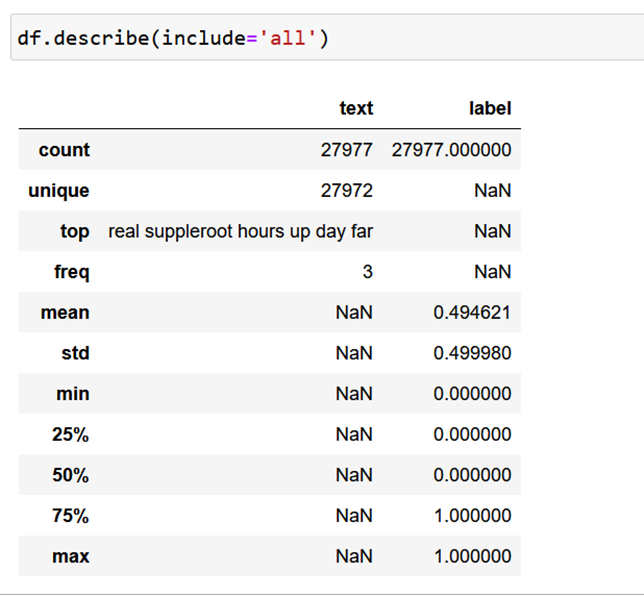
Handling imbalanced data is crucial for ensuring the effectiveness and fairness of machine learning models, especially in tasks such as analyzing mental health with NLP where certain classes or categories may be significantly underrepresented. but in this dataset no need to perform any analysis as we are using data as it is



**Exploratory Data Analysis**

Descriptive statistical

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has  a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

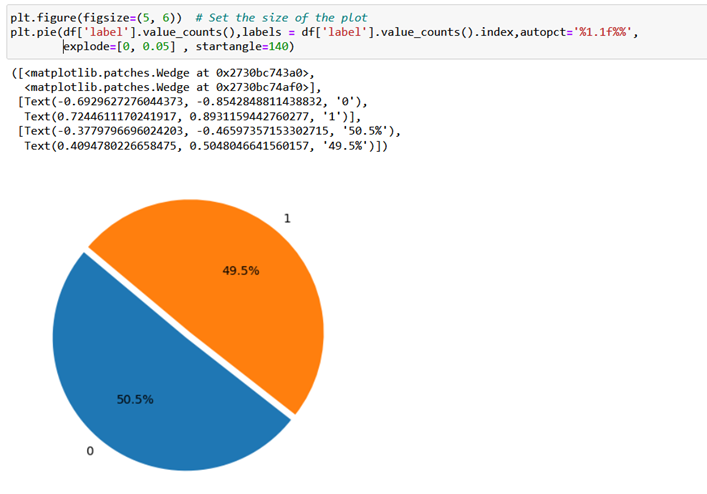


**Visual analysis**

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

**Univariate analysis**

In simple words, univariate analysis is understanding the data with single feature. Here we have displayed two pie chart



**Bivariate analysis**

To find the relation between two features we use bivariate analysis. Here we don’t have 2 numerical columns in our dataset.

**Multivariate analysis**

In simple words, multivariate analysis is to find the relation between multiple features. Here we don’t have multiple features in our dataset.

**Text pre-processing (python packages)**

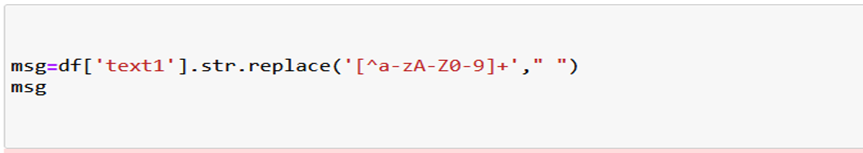
Text pre-processing is a crucial step in Natural Language Processing (NLP) and Information Retrieval (IR) tasks. The goal is to convert raw text into a more meaningful and manageable representation for further analysis.

In Python, there are several packages that provide support for text pre-processing operations. Some of the most common ones are:

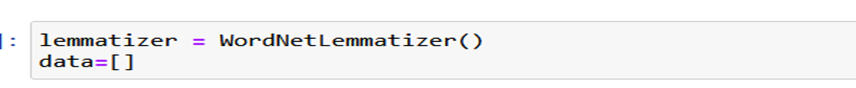
NLTK (Natural Language Toolkit)-It is one of the most widely used NLP libraries in Python.   It provides tools for tokenization, stemming, lemmatization, stop-word removal, and more.



Removing Stopwords–Removing stopwords is a common preprocessing step in natural language processing (NLP) tasks, including analyzing mental health with NLP. Stopwords are common words that occur frequently in a language but typically do not carry much semantic meaning. Examples of stopwords include "the," "is," "and," "in," etc



This will replace all occurrences of alphanumeric characters, plus signs, and double quotes in the 'text1' column with spaces.

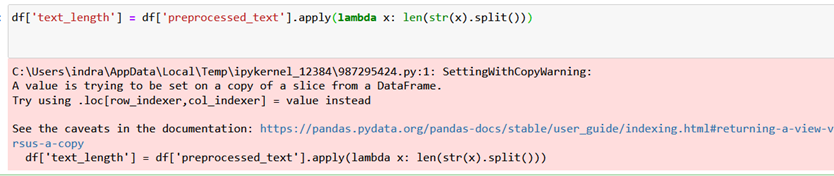
WordNet Lemmatizer is a linguistic tool used to normalize words to their base or dictionary form, known as the lemma. It operates by mapping words to their root forms, helping to reduce inflected forms to a common base for analysis or comparison.



This code we are creating a function that will preprocess the text in the 'text' column of the DataFrame by tokenizing each sentence, lemmatizing the tokens, and then joining the lemmatized tokens back into a single string



TFIDF Vectorizer: TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure used in natural language processing and information retrieval to evaluate the importance of a word in a document relative to a collection of documents

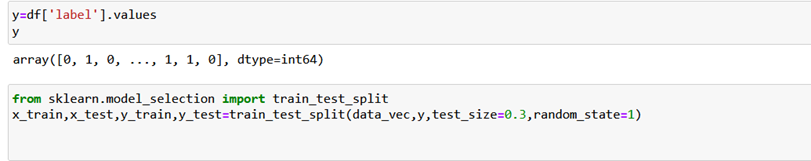


This code will add a new column 'text\_length' to the DataFrame 'df', containing the length of each text in the 'text' column.

After Text Pre-Processing:-

After text pre-processing, the data will typically look much cleaner and more manageable. The specific form it takes will depend on the specific operations performed during pre-processing and here is b

f is passed with dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using train\_test\_split() function from sklearn. As parameters, we are passing x, y, test size, random state.



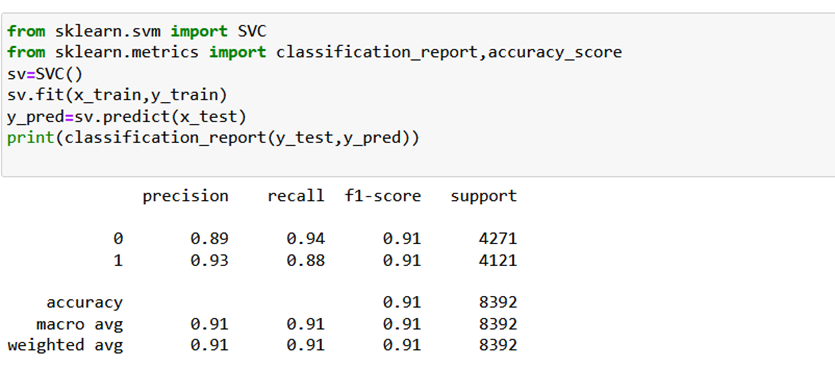
**Model Building**

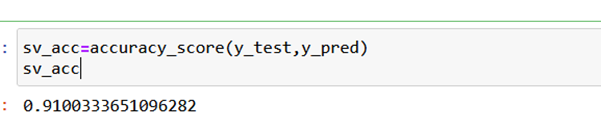
Training the model in multiple algorithms

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms.

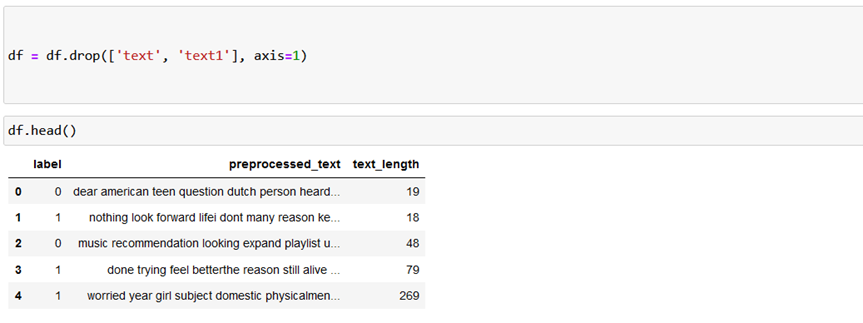
**Support Vector Classifier**

A function named Support Vector Classifier is created and train and test data are passed as the parameters. Inside the function, Support Vector Classifier algorithm is initialized and training data is passed to the model with the. fit () function. Test data is predicted with. predict () function and saved in a new variable. To Evaluate the model Classification report is used.





elow fig after cleaning the text



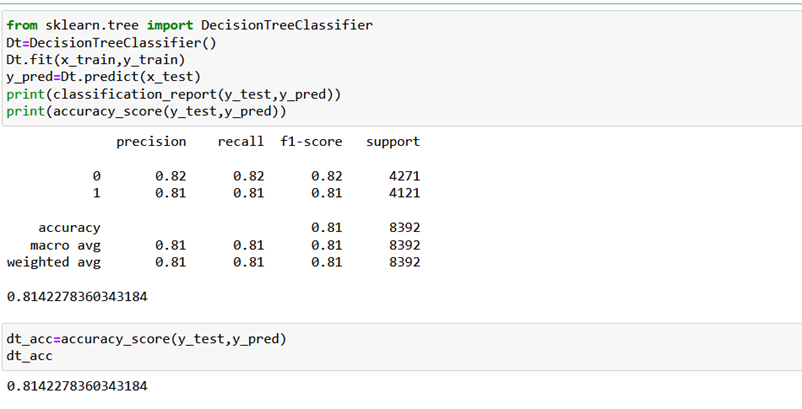
Splitting data into train and test

Now let’s split the Dataset into train and test sets. First split the dataset into x and y and then split the data set

Here x and y variables are created. On x variable, d

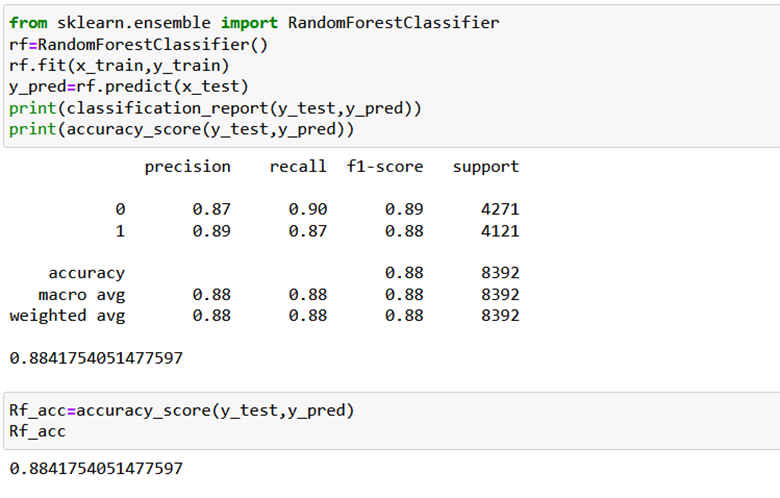
**Decision Tree Classifier**

A function named Decision Tree Classifier is created and  train and test data are passed as the parameters. Inside the function, Decision Tree Classifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. To Evaluate the model Classification report is used.



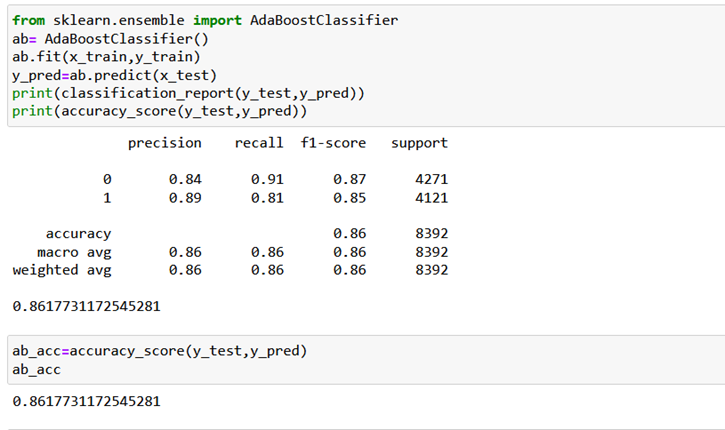
**Random Forest Classifier**

A function named Random Forest Classifier  is created and  train and test data are passed as the parameters. Inside the function, Random Forest Classifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. To Evaluate the model Classification report is used.



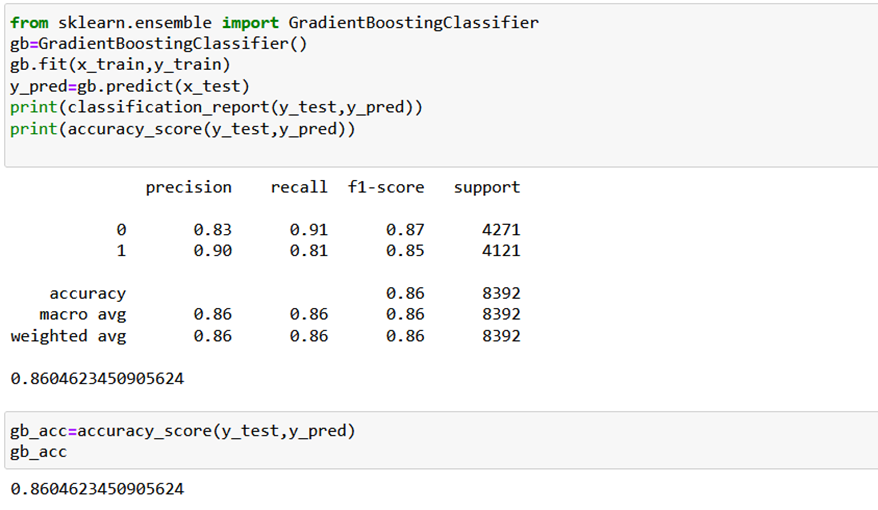
**Adaboost Classifier**

A function named Adaboost Classifier  is created and  train and test data are passed as the parameters. Inside the function, Adaboost Classifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. To Evaluate the model Classification report is used.



**Gradient Boosting Classifier**

A function named Gradient Boosting Classifier is created and  train and test data are passed as the parameters. Inside the function Gradient Boosting Classifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with. predict () function and saved in a new variable. To Evaluate the model Classification report is used.



**Logistic Regression**

A function named Logistic Regression  is created and  train and test data are passed as the parameters. Inside the function Logistic Regression algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. To Evaluate the model Classification report is used.



**Performance testing and Hyper Parameter Tunning**

Testing the model

Compare the models

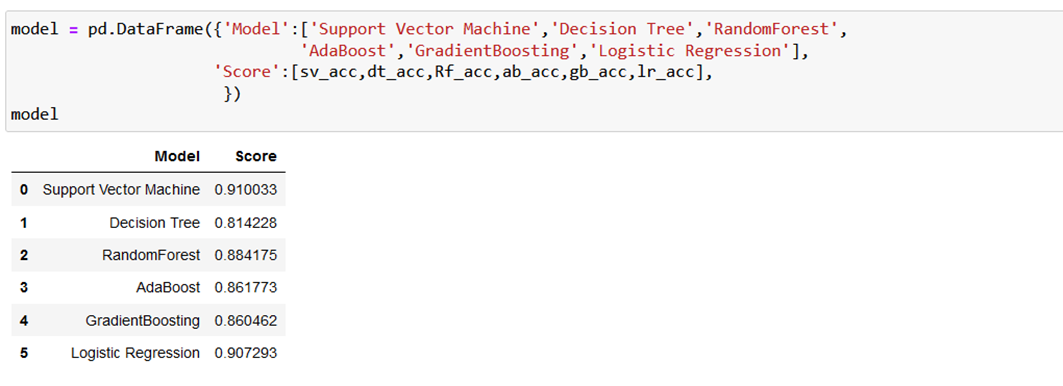
For comparing the  models we create data frame with accuracy of models

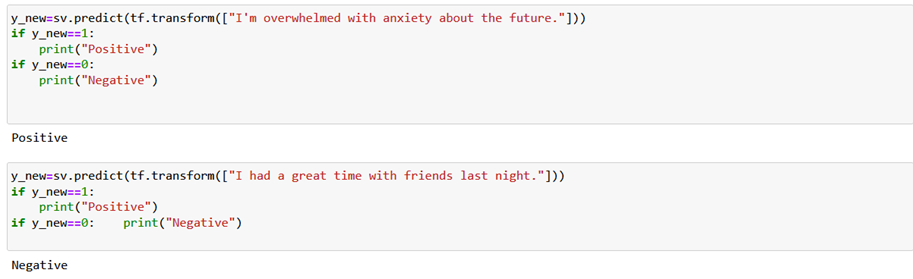
Testing the model

Since form above comparison Support Vector Machine is performing well we will test for that model.

**Activity 2:** Comparing model accuracy before & after applying hyperparameter tuning (Hyperparameter tuning is optional. For this project it is not required.)

Note :- We are Getting Good Accuracy with Support Vector Machine so, no need of Hyperparameter Tuning



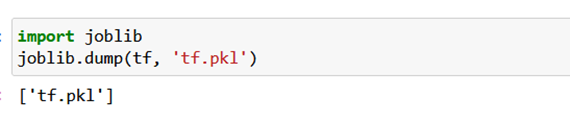


**Model Deployment**

Save the best model

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.





**Integrate with Web Framework**

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI. This section has the following tasks

* Building HTML Pages
* Building server-side script
* Run the web application

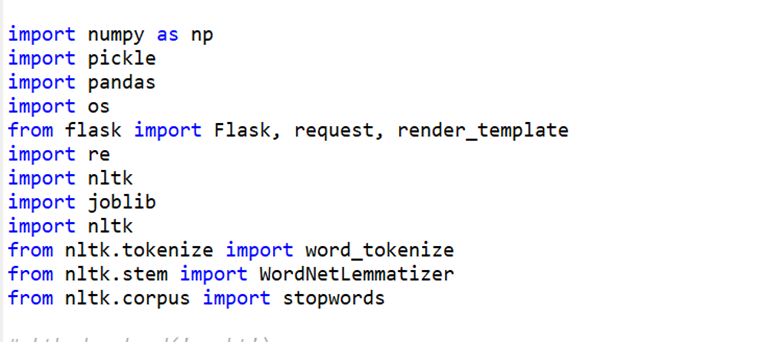
**Building Html Pages:**

For this project create two HTML files namely

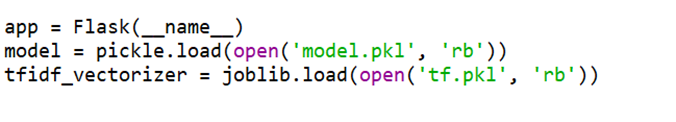
* Index.html
* Input.html
* Output.html

and save them in the templates folder.

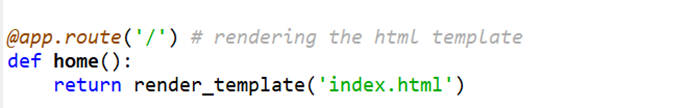
**Build Python code**



Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module ( name   ) as argument.

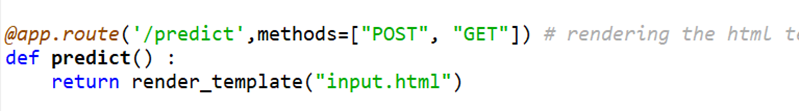


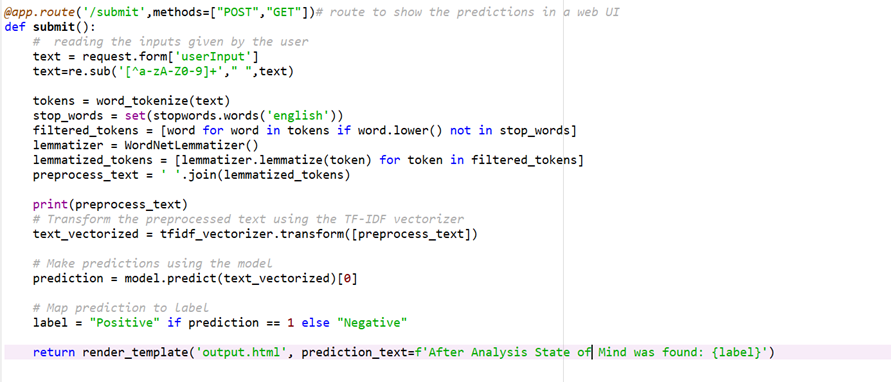
Render HTML page:



* Here we will be using a declared constructor to route to the HTML page which we have created earlier.
* In the above example, ‘/’ URL is bound with the index.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered.
* Whenever you enter the values from the html page the values can be retrieved using POST Method.

       Retrieves the value from UI:





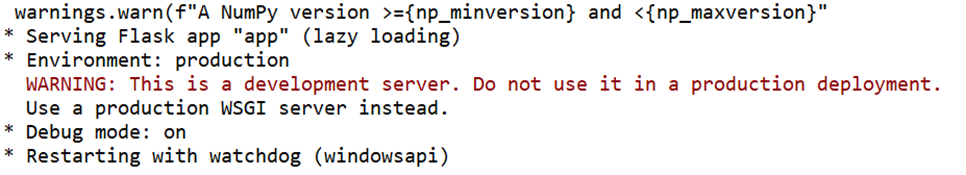
 Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the input.html page earlier.

Main Function:

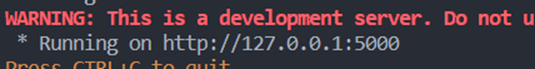


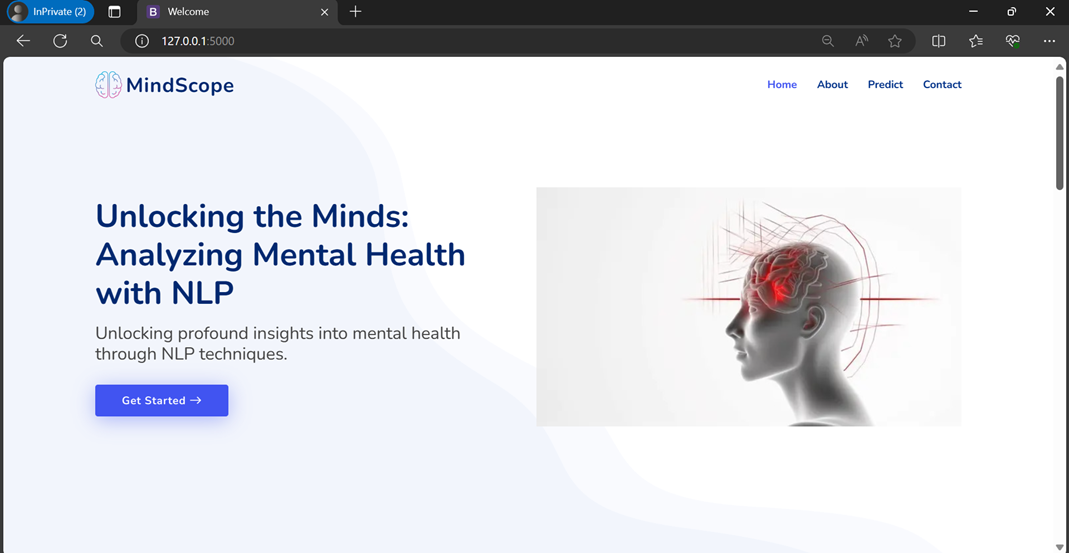
**Run the web application**

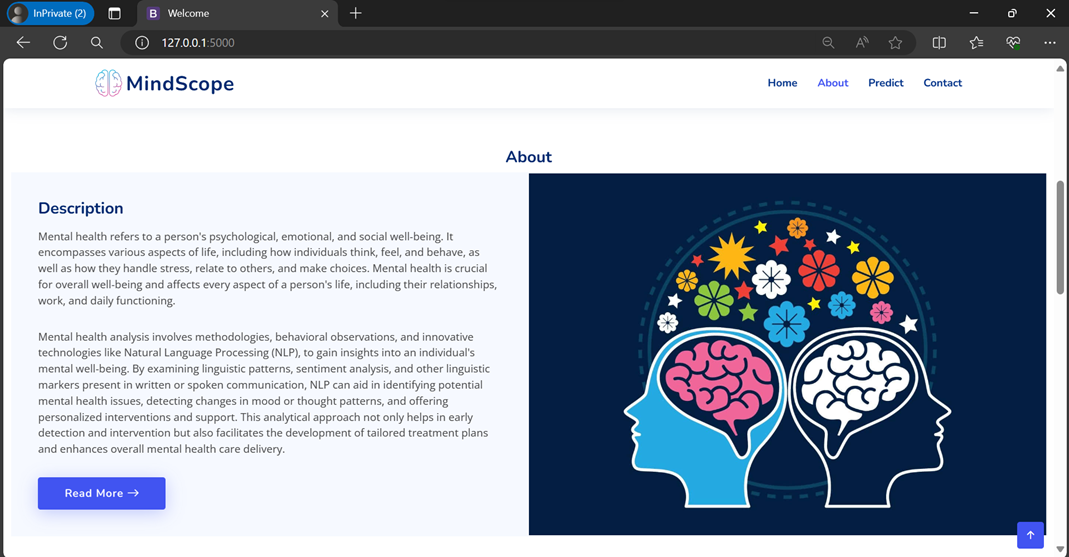
* Open anaconda prompt from the start menu
* Navigate to the folder where your python script is.
* Now type “app.py” command
* Navigate to the localhost (127.0.0.1:5000) where you can view your web page.
* Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

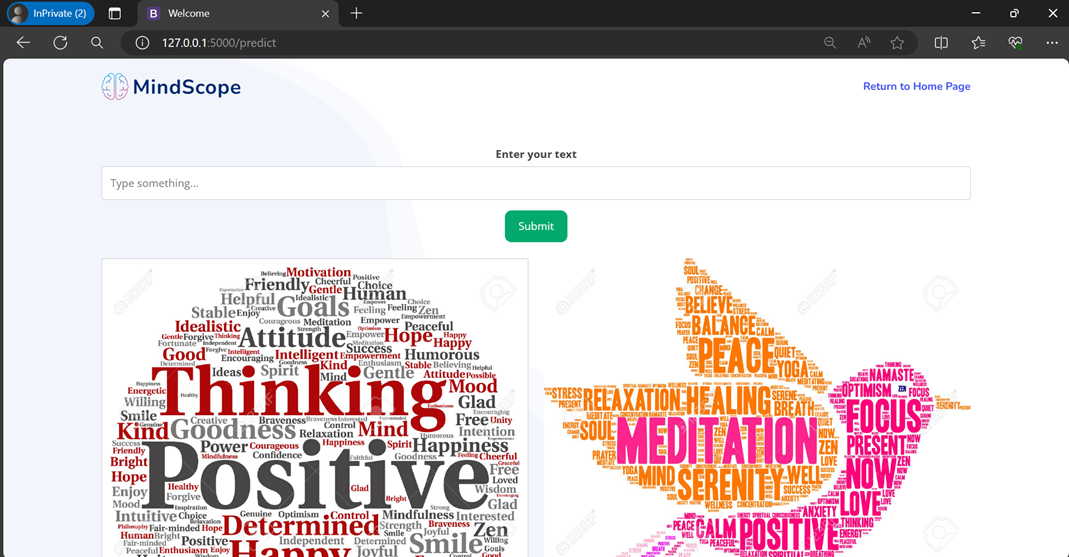


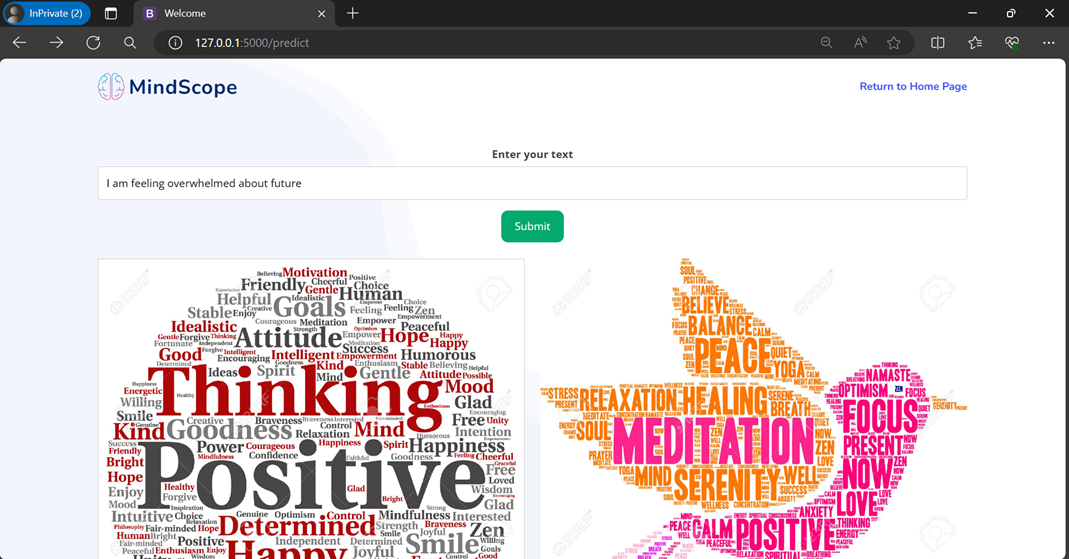
If you set debug as False you can see the port address as below

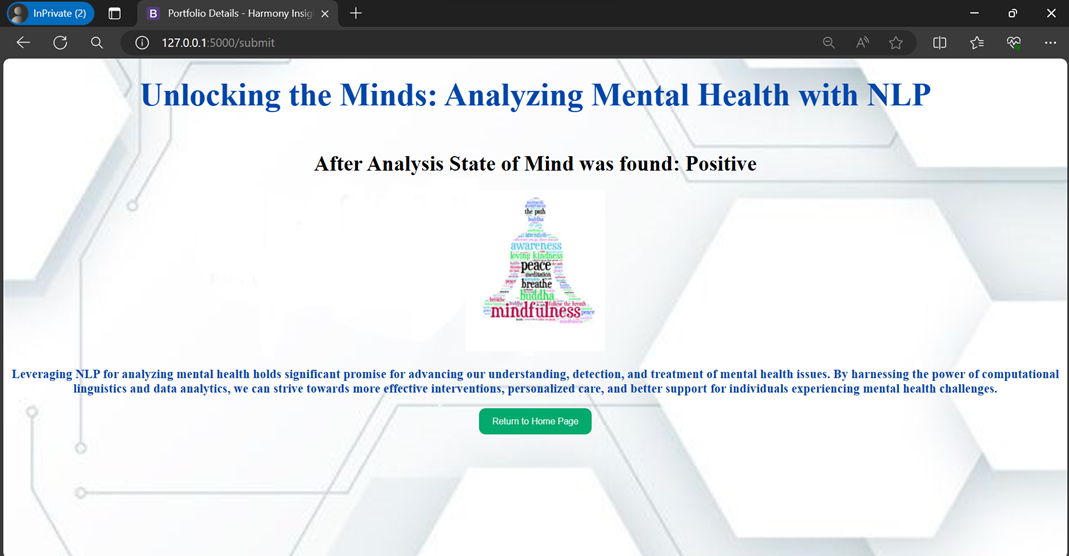


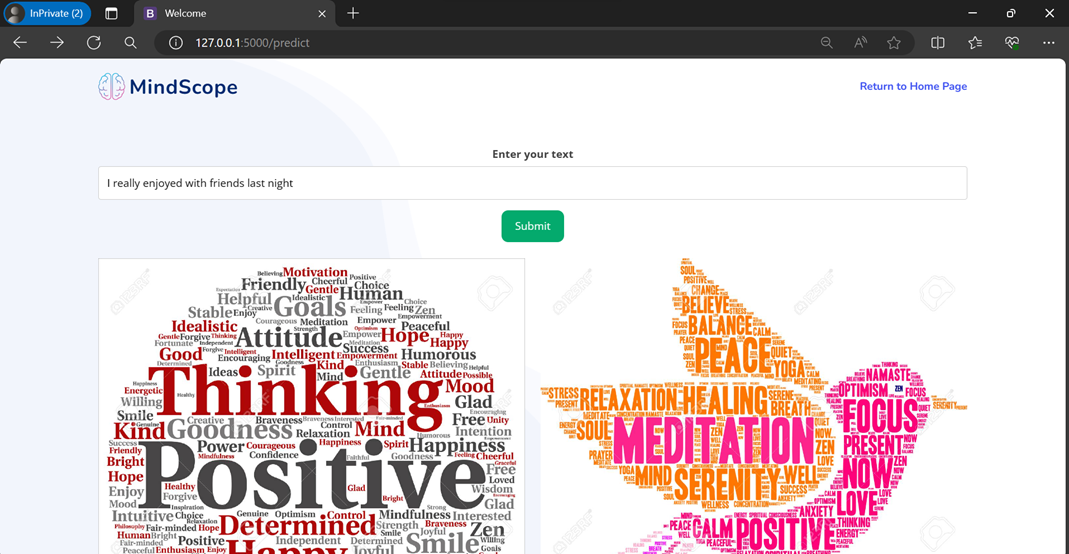


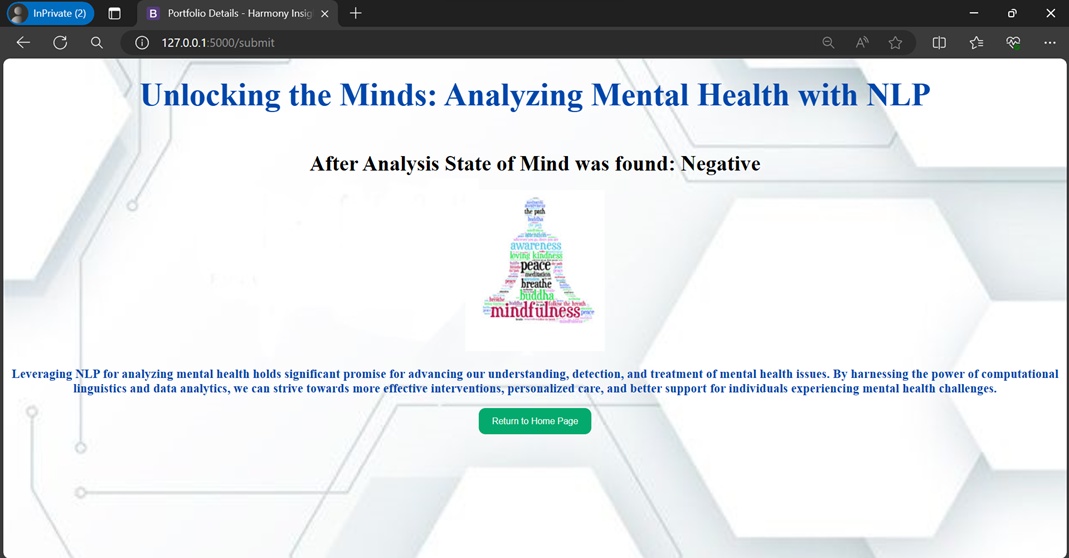












**CONCLUSION:**

The intersection of Natural Language Processing (NLP) and mental health offers a transformative opportunity to understand and address mental well-being more effectively. NLP’s ability to process vast amounts of textual data enables early detection, continuous monitoring, and personalized interventions for mental health challenges. Through the analysis of patterns in language, sentiment, and even conversational nuances, NLP tools can provide insights that were once difficult to obtain.