1. The project we are working on relates to mental health issues. We will detect emotions for the posted text data and then will suggest some motivational quotes, related to that emotion.

So, the target group of our mini-project includes social media users, mental health advocates, and general users. Social media users may receive mental support as a recommendation, after analyzing their social media posts. Mental health advocates may use this one as automated support for his or her clients when needed urgently.

End users are those people, who will interact with the application to get personalized quotes. Maybe they will seek motivational, comforting, or inspirational content, based on their current mood and mental state. For the end users, our application must detect the emotion with possible maximum accuracy. Our system must recommend relevant quotes. User privacy should be maintained when we scrap social media data. Our project will offer scalability as multiple features can be added later and existing features can be updated in more advanced ways.

All of the objectives, of this project, may not be possible to achieve after completing this mini-project. But we want to have our users, emotional support, self-reflection, and personalization. Our users will have mental support when needed. We will also try to provide updates on their mental state based on their social media updates and personalized contents.

There will be multiple benefits to the proposed project solution. Users will have recommendations based on their profile which is not pre-defined. If someone is concerned of someone's mental state, he or she will have the opportunity to always be updated. Users will have comfort as well as motivation from the recommendations. In future iterations, users will have more customized recommendations, based on their feedbacks.

1. We will have three types of data sources.

To train our model for classifying the text emotions, we searched data set both in Kaggle and GitHub repositories. Fortunately, we have found multiple datasets. So, need to decide, which one is more applicable for training our model. Then, we are also looking for a data set of motivational quotes with proper labeling.

Then, initially, we will collect a single text from the user using an web application to detect the emotion. But we also planning to use social media APIs to scrap enough data from a single user’s profile to analyze both the texts and the mental state of the user.

Now, we will use only a static dataset to recommend quotes. But in future work, maybe we will do web scrapping to have dynamic quotes and recommendations.

We will have a text dataset that includes the text and labels for classification. Then, we will also have a similar dataset of motivational quotes and related emotional labeling. We will also have a user text table, where we will have a user ID, username, text content, and detected emotion. In another user table, we will have user ID, username and quote preferences.

In future we will try to keep a feedback table with user id, quote id, rating and comments.

For data collection, initially we will use pre-defined datasets from Kaggle and Gihub. User text will be collected and analyzed using a flask web application. In the future, we will use APIs to scrap social media data.

We will use a .csv file to store our training data and another one for quote data. We may use a relational database to store user and user feedback-related information.

For data preprocessing, we will use text preprocessing techniques like, lowercase letter conversion, stop words removal, and stemming. For these, we will use nltk library. We may also use other libraries. We use classification algorithms to classify emotions.

We are still not concerned about data privacy and security, as we are not dealing with sensitive information. But in the future when we will use social media data, we will ensure that user data will be handled securely and we may also implement access control mechanisms.

1. We have multiple goals related to data preprocessing.

We will clean and standardize our data to improve data quality. It will improve the model's performance. We will try to eliminate noise from our dataset to avoid overfitting and underfitting problems.

Data preprocessing may include, text conversion to lowercase and punctuation removal. Then we will tokenize the text. Stop words will be removed as a next step. Then stemming or lemmatization will be performed to have the basic form of tokens. We will use either text vectorization or TF-IDF calculation.

For the data cleaning part, we will try to remove duplicate entries so that, it doesn’t bias our model.

For data transformation, maybe we will use word embedding by Word2Vec to capture semantic meanings. We may also use BERT to identify word contexts. We are planning to use parts of speech or POS tags to include grammatical tags.

For the feature engineering, we will assign single emotion per text. But we trying assign multiple emotions to a single one, to extract detailed ideas behind one post. We will work with text length. If it is higher than the expected length, we will use the text rank algorithm to summarize it.

1. To explore and understand the data, we will examine a subset of data, as it is not possible to each sample separately. We will try to identify the most frequent words using multiple techniques. Maybe we will use a word cloud. We will try to visualize emotion distribution to eliminate over-sampling issues using histogram or box plot. We will analyze term frequencies, to identify mostly used but unnecessary words in the dataset.

To will consider text length to decide on summarization. We will work with missing values. If any label is missing, we cluster the emotions and assign a label to the missing one. We may work with the most significant terms with each topic or emotion.

1. We will use a word cloud to visualize the most frequent words. We will try to plot an emotion distribution chart to show the frequency of different emotions using a pie chart or bar chart. We will try to analyze text length, using histogram. To show the cluster of different texts, we may use scatter plot. If we use the feedback on quotes feature, we may also demonstrate it using a list format.

Similarly, we may use an interactive word cloud, to find out the most frequent words posted by the users. Interactive topic exploration may enable users to click on different topics. Sentiment trend analysis may engage a user to find out tending emotions in a certain period.

We are using Python programming language to build our web application. So different Python libraries like \_\_\_\_ we will try to incorporate for this visualization part.

1. If we want to define our problem, we do it in three categories. The supervised machine learning algorithm to classify the text emotions. Then, we will build a recommendation system implementing unsupervised machine learning algorithm. So, this one is clustering algorithm. And then finally, we try to build a small web application to use the machine learning models. We are planning to build and Flask app.

So, we will learn, how to do the emotion classification, and how to cluster texts for recommendations and then, we HTML, CSS, and JavaScript to develop the app.

Input variables will be tokens, text length, and sentiment labels, collected from our training data set, with user text posts. And the choice of recommendation. We may also consider some historical data like previous posts, and feedback from the user.

1. We have not decided our ML and statistical methods. For the classification purpose, we will use Naïve Bayes which widely used for text classification. We may also Support Vector Machine which is also common to classify texts after tokenization. For the cluster part, we will use k-nearest neighbor algorithm. For the recommendation, we are planning to use in future collaborative filtering, if we can take input from the user.

To evaluate clustering algorithms, we will use the accuracy rate and confusion matrix to evaluate our model. For the clustering model, we may use the elbow method.

We split our dataset into train and test subsets. We surely implement K-fold cross-validation while training the model to solve the overfitting problem.

1. We will use a web application as the deliverable option to the end user. Web application is highly interactive and accessible. It will allow the users to input their text and personalized motivational quotes.

In the home page, there will be a text box where user can give their inputs. Then a button to submit the post. Then we will have two sections. In one section, the user will see the analyzed emotion from the text. And in another section, there will be quotes. We will try to make the web application as user-friendly and attractive as possible.

1. Users enter their status or message into a text input field on the web app.
2. The system analyzes the text to detect the emotional state using pre-trained models (e.g., SVM, Naive Bayes, or neural networks).
3. Based on the detected emotion, the system retrieves and displays relevant quotes from the database.
4. Show visualizations such as emotion distribution, sentiment analysis, and historical trends.

Future feature workflow

1. Allow users to explore data related to their emotional state and quotes.
2. Provide options for users to rate quotes and offer feedback on recommendations.

System Updates: Use feedback to continuously improve the recommendation engine and adjust the model as needed.

1. Option to analyze social media posts (with user consent) to provide more personalized recommendations and insights.
2. We are working on the built-in dataset. So, there is possibility of bias towards specific groups based on age, gender, or ethnicity. If we work on this project in the future, we will try to use our dataset to avoid the problem. But we will provide equal treatment to all of our users.

Initially, we are collecting single text from the user. So, no consent is required. But, I guess, when we will work with the social media data, we will have confirmation from the user.

While saving one post, to analyze emotion, there is privacy concern. But to analyze social media, there can be a privacy issue. So, we try to store data as hash value, rather than in plain text.