***Title: A self-help chatbot for dealing with anxiety, isolation***.

***Summary*:**

The project aims to develop a chatbot to aid in dealing with anxiety, isolation, or any other mental health issues as an immediate help. The project is divided into two main components as follows:

1. Emotion Analyzer
2. Response Generation

The current scope of the project focuses on the emotion analysis task.

The task at hand is a classification problem and predicts the emotion of a sentence. The data used for the model is publicly available data obtained from Kaggle.

The files contain twitter data and labelled sentences. The sentences span a total of 15 categories of emotions as shown in the figure below.

Chart, bar chart

Description automatically generated

*Methods*:

The sentences are undergo the following workflow:

1. Pre-processing: Data cleaning involves removing stopwords, punctuations, links, usernames and extra spaces. This data comprising of sentences are divided into tokens and each token is converted to its root lemma using ‘nltk’ packages in python.
2. Vocabulary generation: The unique words found in the data are collected in the vocabulary.
3. Converting text sequences into numerical values: The text sequences need to be converted into a numerical format which is done using the ‘keras.tokenizer’. Since the length of the sequences can be variable, the length of all the numerical sequences are standardized and shorter sequences are padded with zeros.
4. Glove embeddings: The vectors for the sequences are represented as vectors for each word being represented in 100D using the Glove pre-trained word embeddings. This helps to find the relationships between similar words.
5. Model development: The data is divided into training and test datasets with test data accounting for 25% of the total data and 20% of the training data is used as validation data. The data is used for training 3 types of models: Recurrent Neural Networks (RNNs), Long Short Term Memory(LSTMs) and Bidirectional Long Short Term Memory(Bi-LSTM)
6. Model Training and evaluation: The models are trained using categorical cross-entropy loss function and Adam optimizer for 25 epochs with a mini-batch size of 128.

*Results*:

1. The Bi-LSTM model shows the best performance for all the models evaluated. With a training accuracy of 61.1%
2. The performance tends to improve by adding Dropout layers and increasing the LSTM layers as observed during training the model.

*Future Scope*:

The model performance can be improved using pretrained models like BERT.

Also, the models can be trained using GPU to reduce the training time.