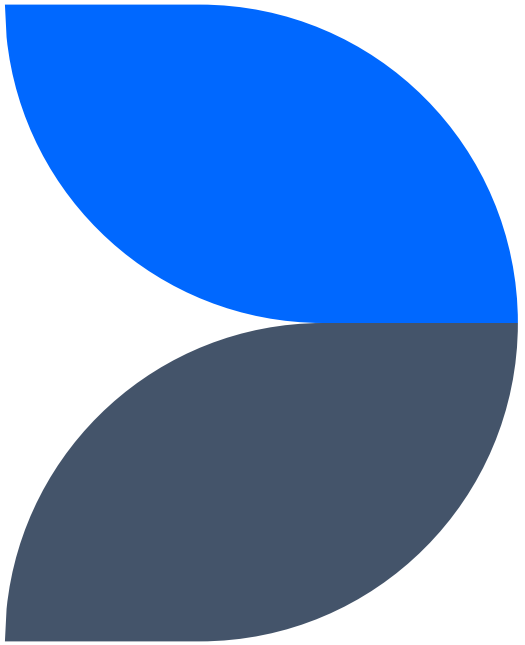
Dataset



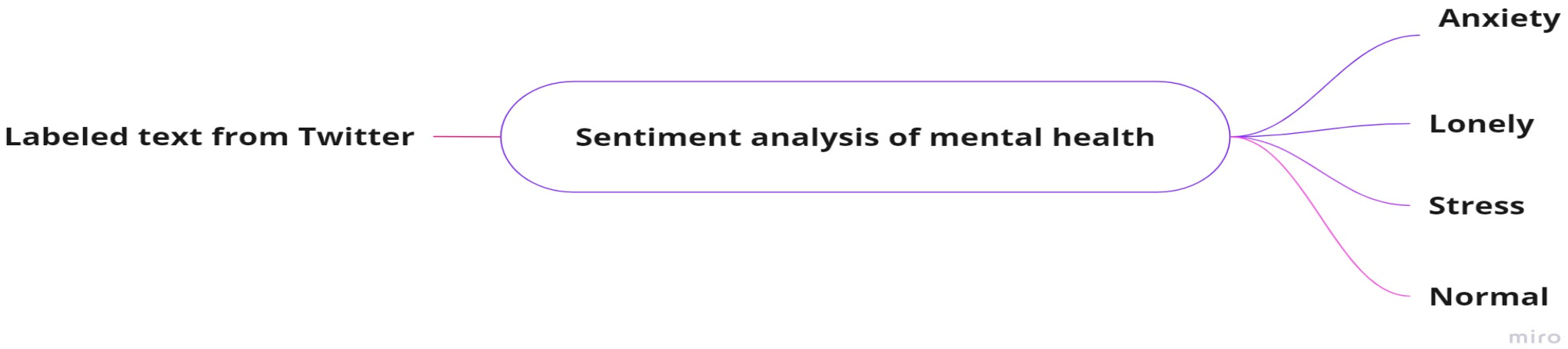
Dataset

- Scraped using Tweepy API for the purpose of working on the idea of tracking Mental Health from Social Media.
- Labeled based on expert studies in the mental health domain.
- Contained from 4 features: stress, lonely, anxiety, and normal.
- Found as cleaned dataset on: https://www.kaggle.com/datasets/arshkandroo/behavioural-tweets, but after contacting “Arsh Kandroo”, he sent me the dataset before cleaning.

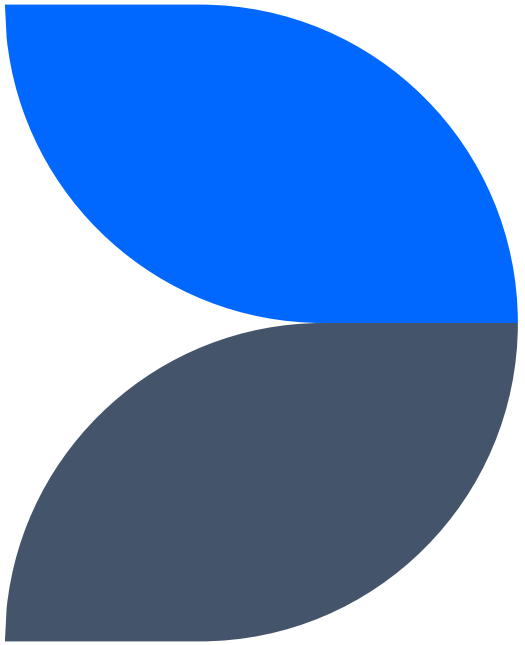
Use case

Use case

- Classify the input text into its class from the 4 classes we have to know the sentiment of the input.



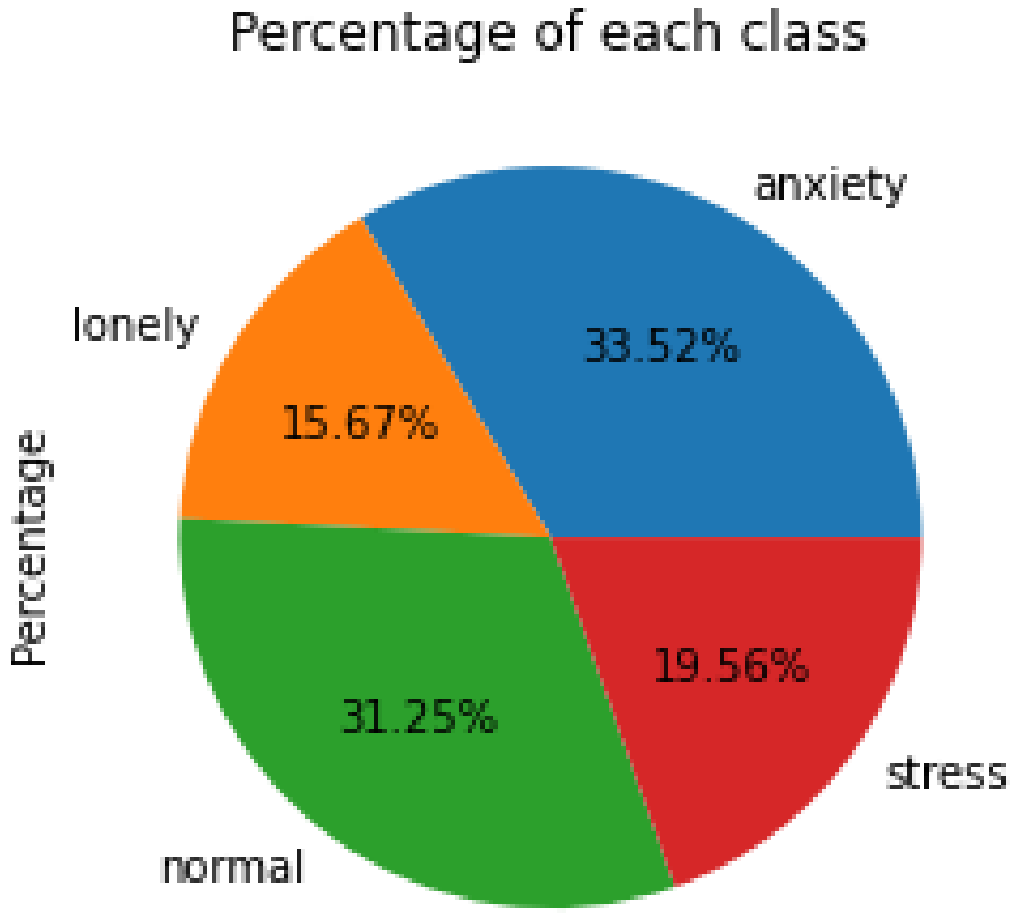
Data Exploration



Data Exploration

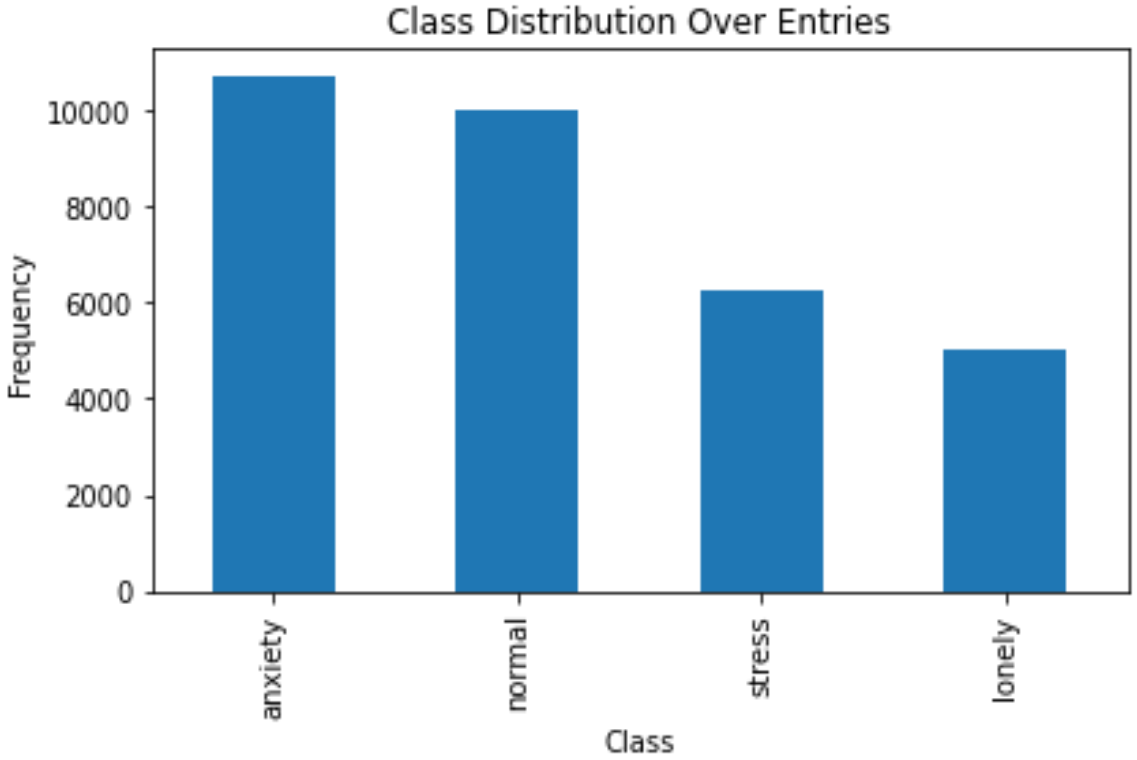
- Visualize the percentage of each class in the dataset using pie chart
- Visualize the distribution of the classes vs the frequency of them
- Visualize the most important words in each class using Word Cloud library
- Visualize the top countries that have mental health problems
- Visualize the days of the week to show when people suffer from mental health problems.

Data Exploration



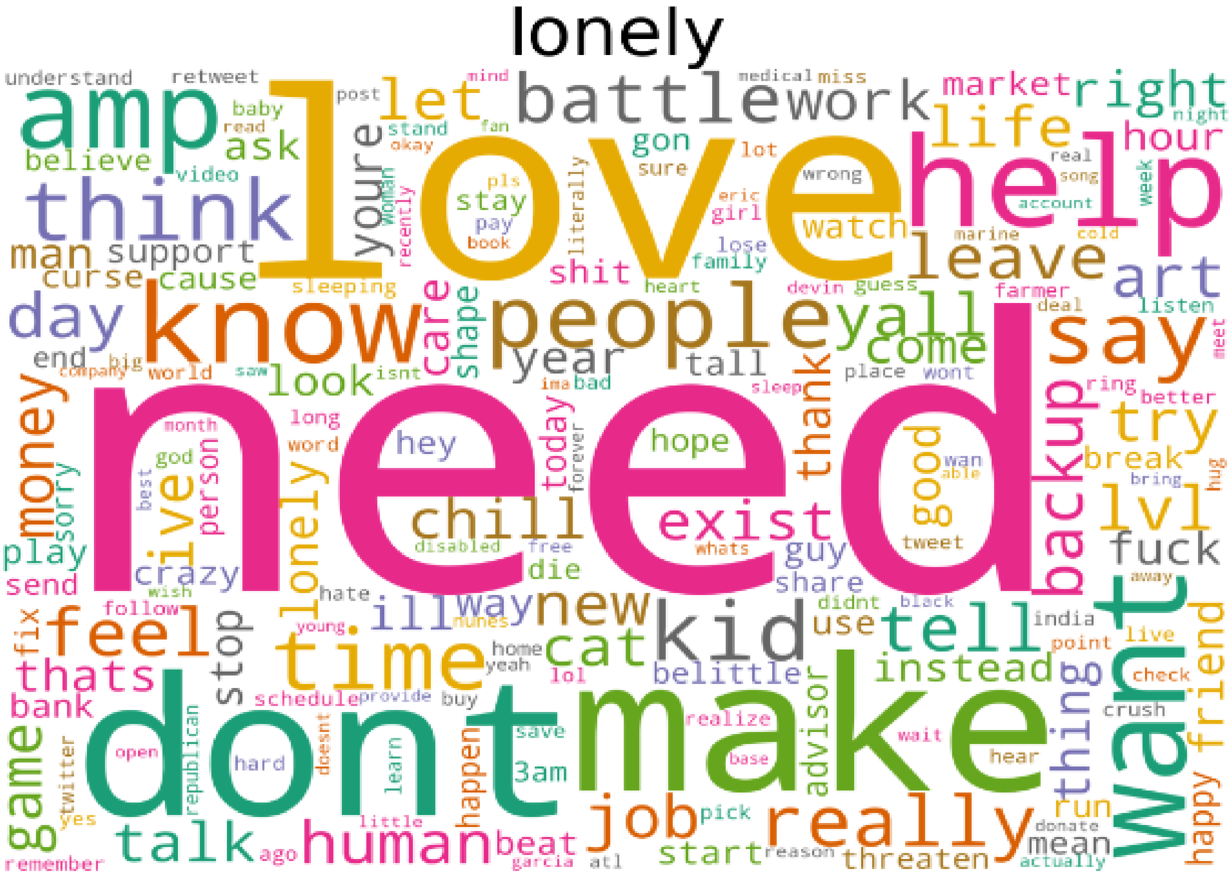
The percentage of each class

Data Exploration



The distribution of the classes

Data Exploration

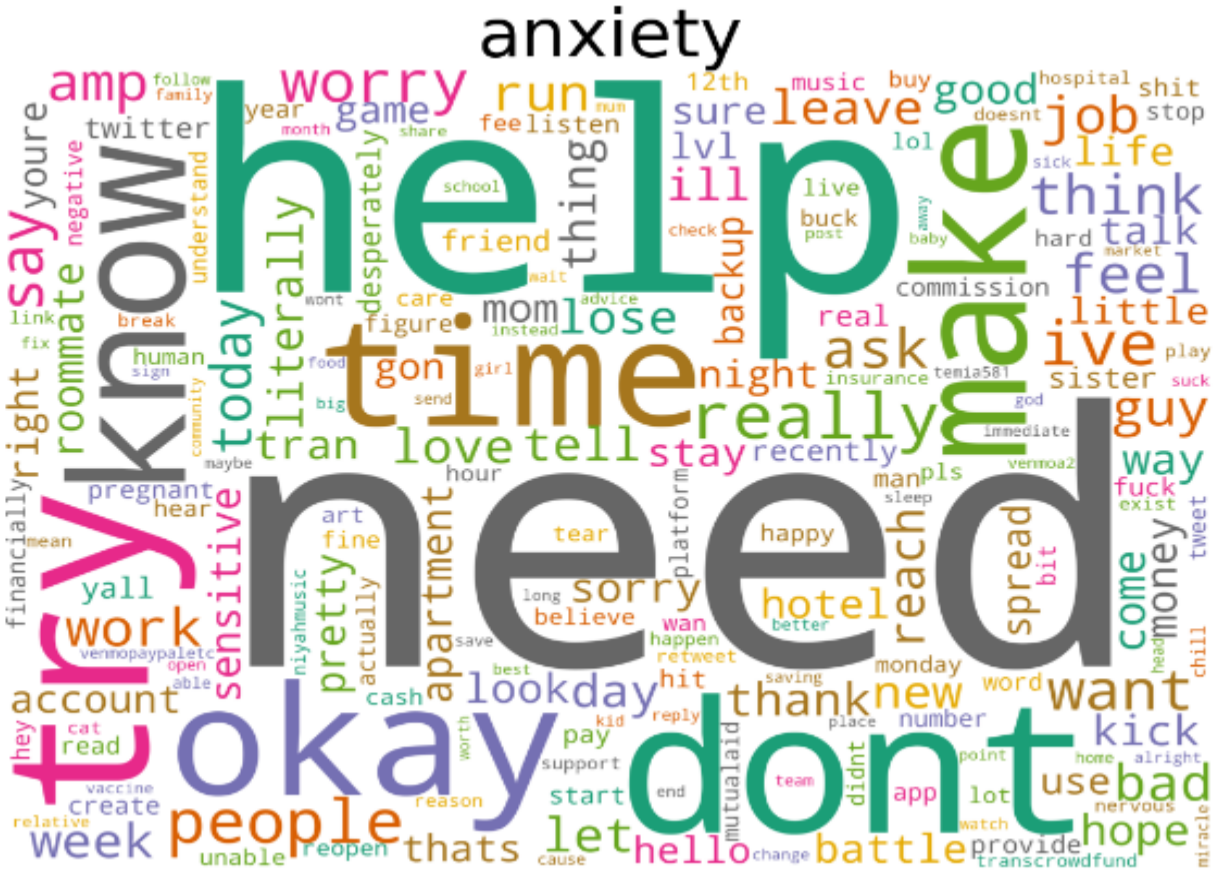


The most important words in the lonely class

Data Exploration

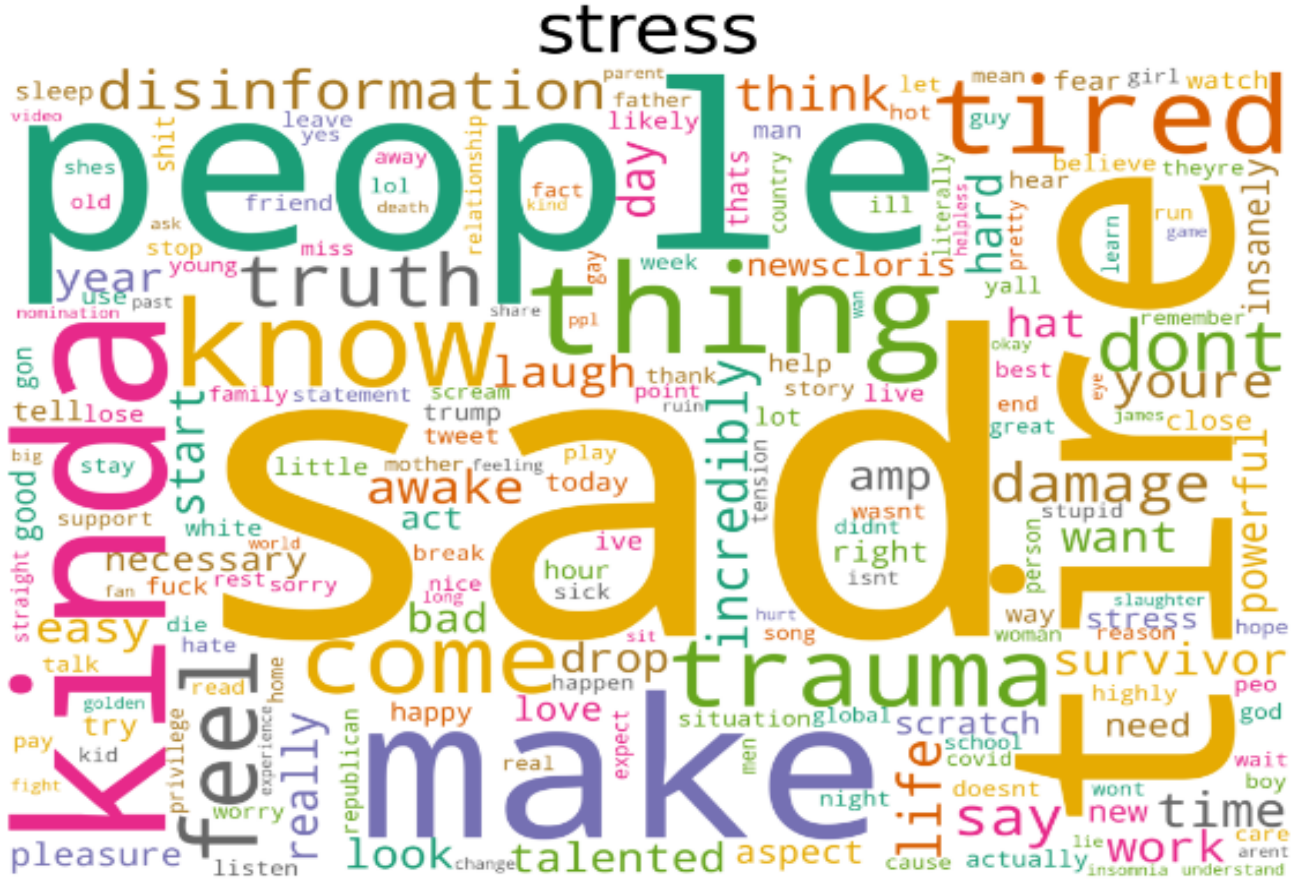
The most important words in the normal class

Data Exploration



The most important words in the anxiety class

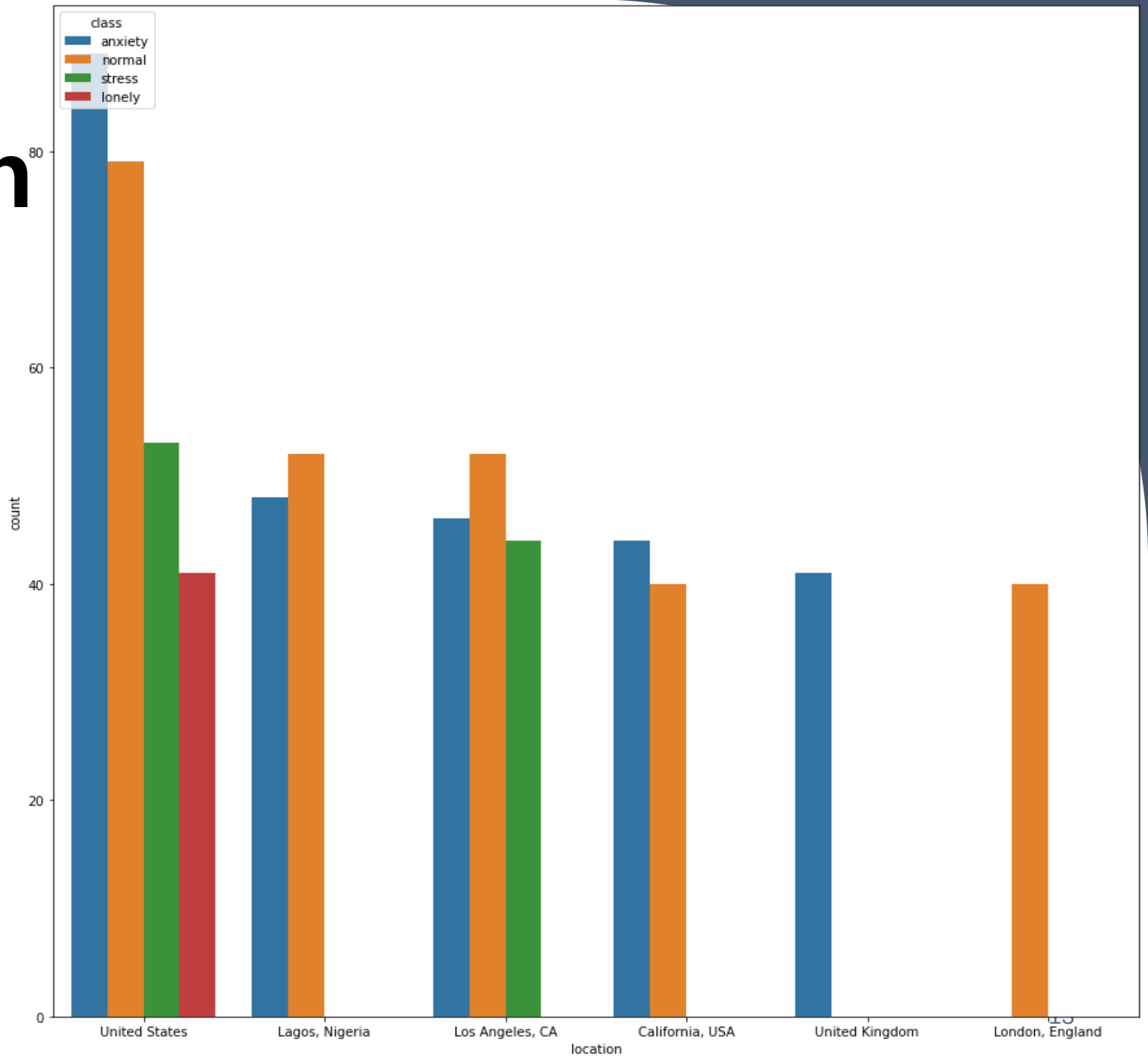
Data Exploration



The most important words in the stress class

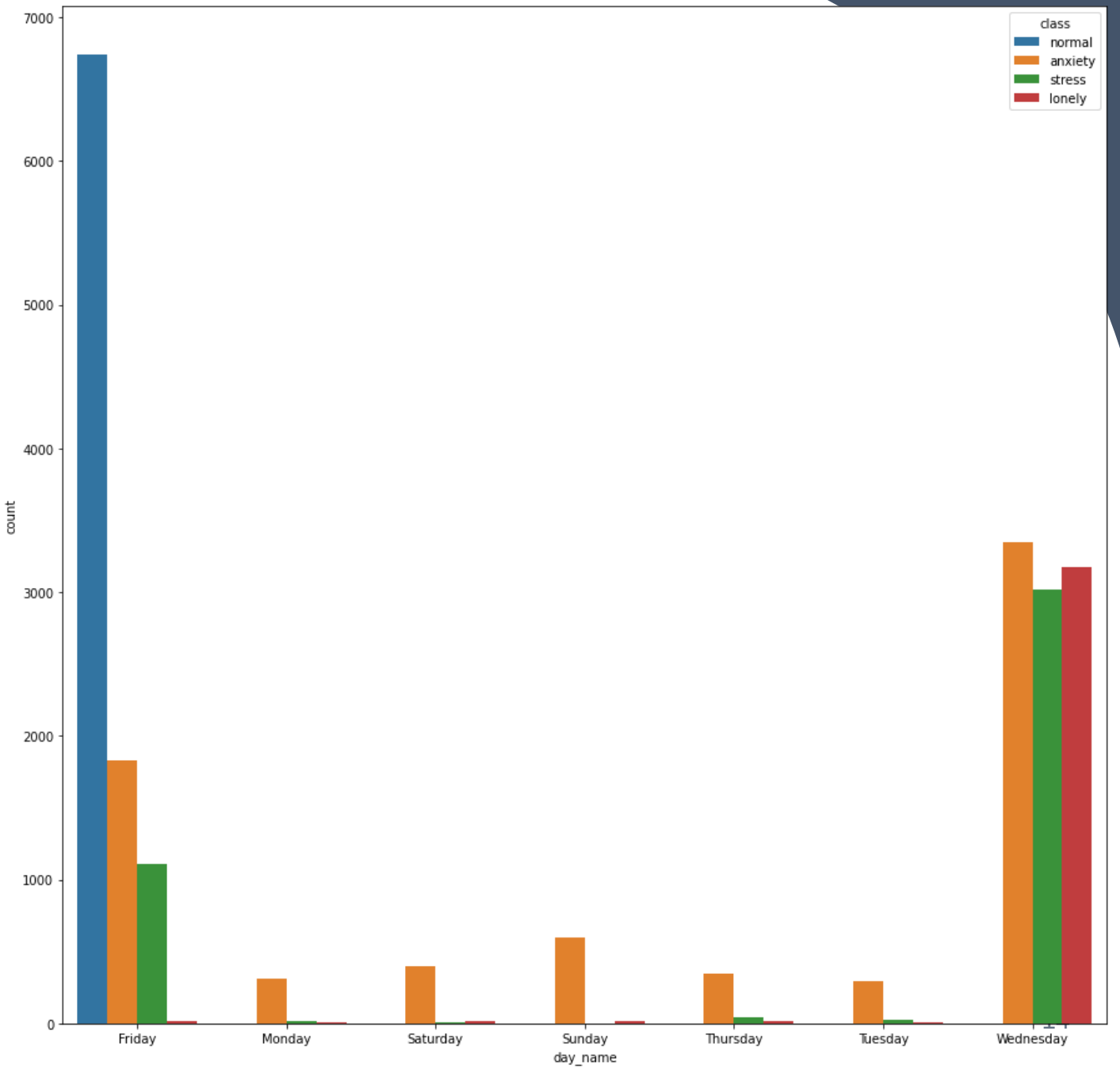
Data Exploration

The top countries that have mental health problems

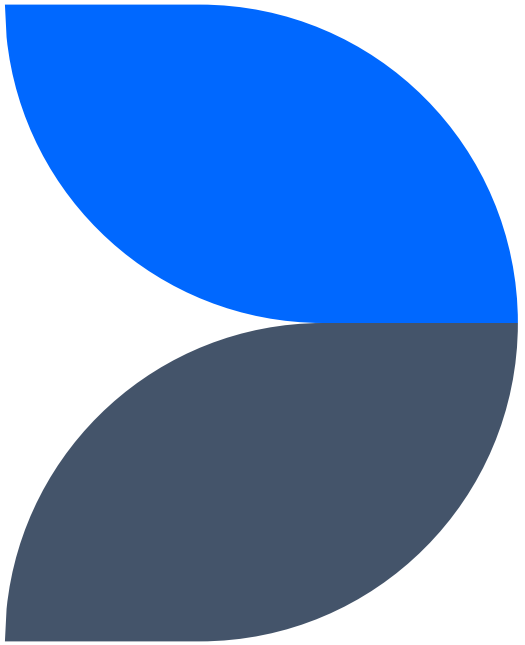


Data Exploration

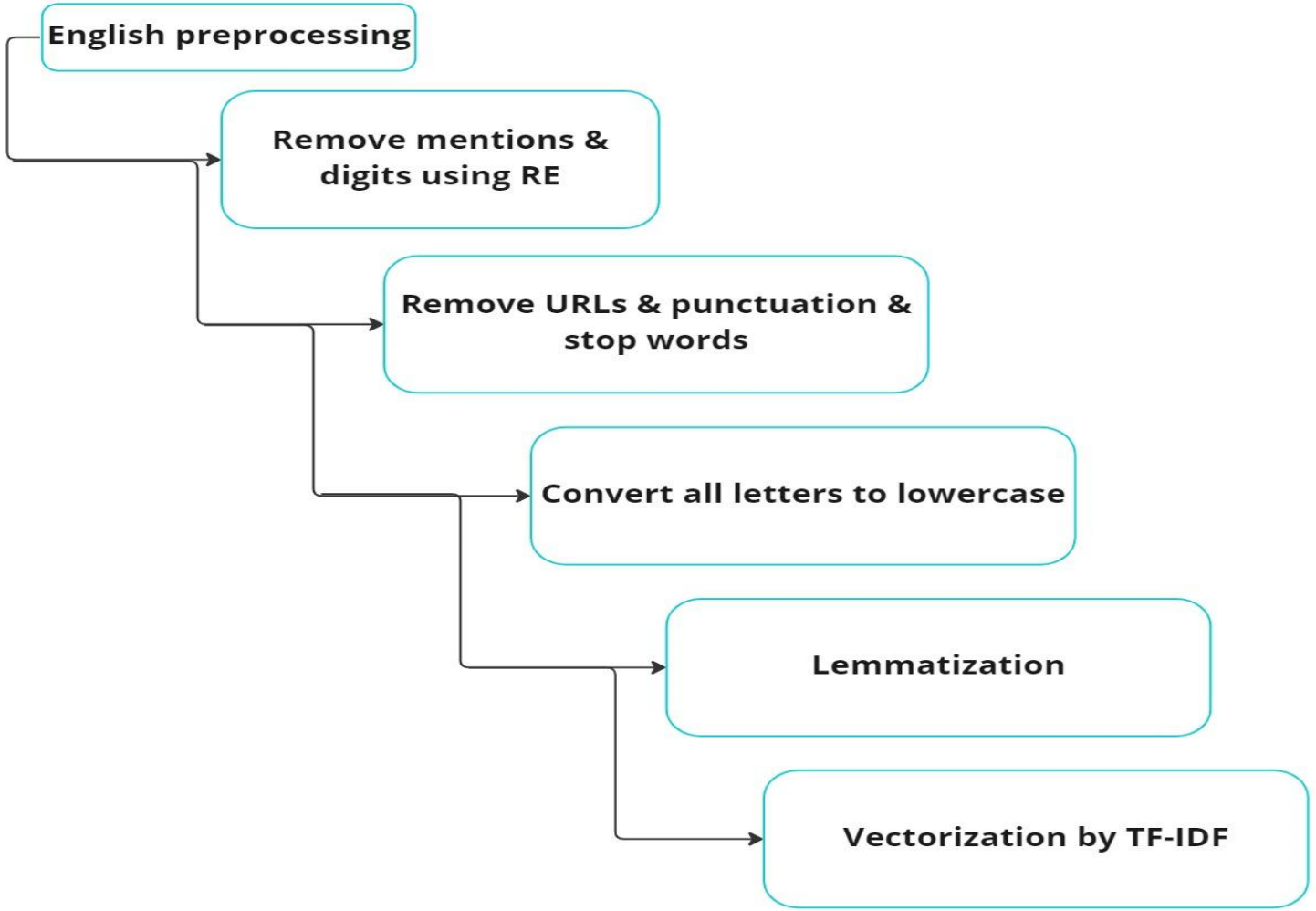
Days of the week that people suffer from mental health problems



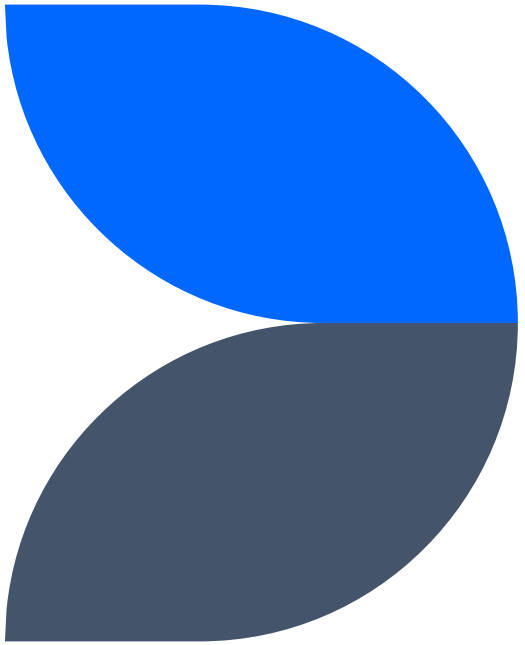
Data Preprocessing



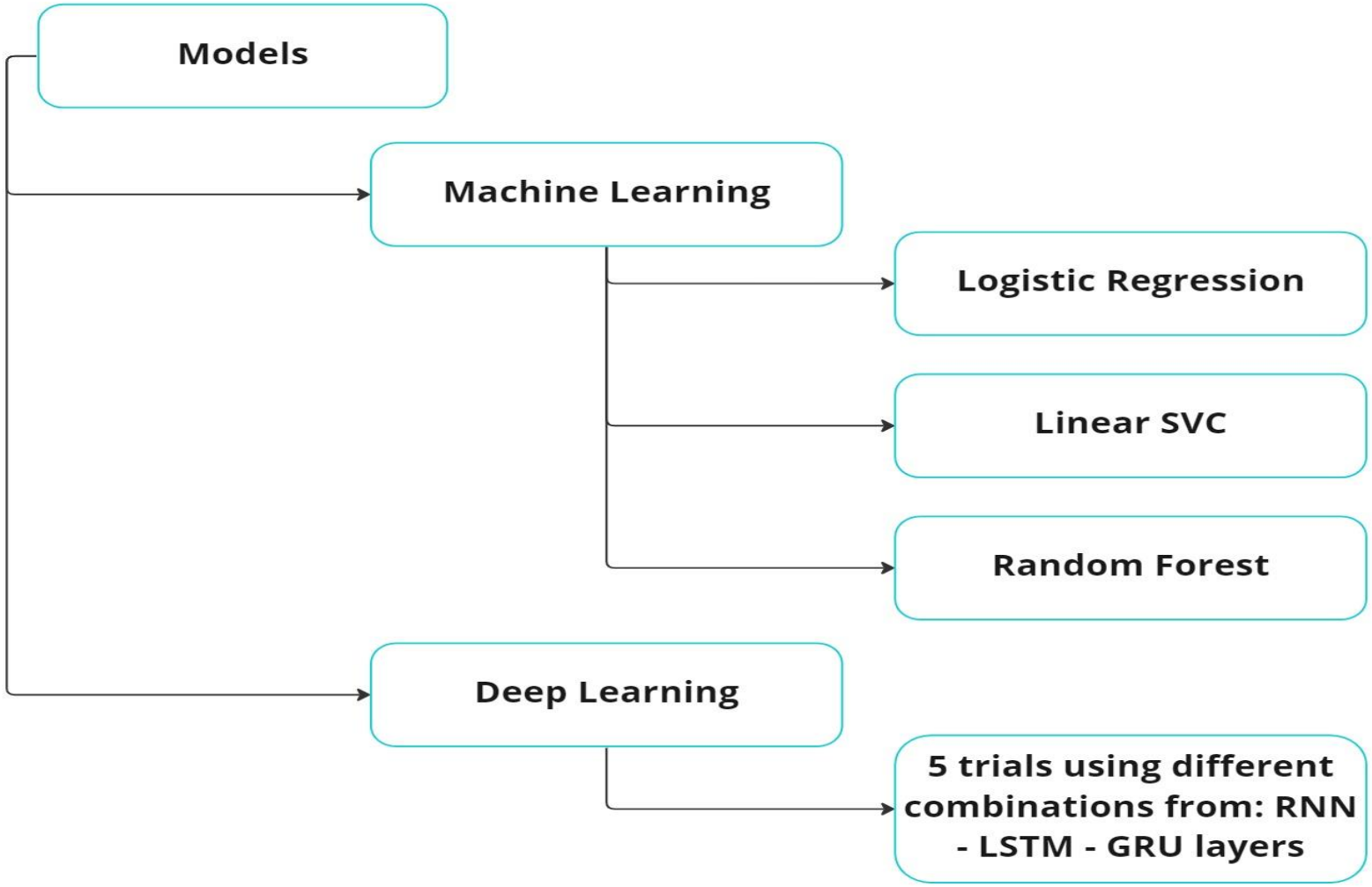
Data Preprocessing



Data Modelling



Data Modeling



Results

Results (Machine Learning)

Logistic Regression

Accuracy: 74.98%

Classification report:

	precision	recall	f1-score	support
anxiety	0.72	0.72	0.72	3260
lonely	0.46	0.34	0.39	1495
normal	0.77	0.90	0.83	3045
stress	0.93	0.88	0.91	1801
accuracy			0.75	9601
macro avg	0.72	0.71	0.71	9601
weighted avg	0.74	0.75	0.74	9601

Linear SVC

Accuracy: 73.81%

Classification report:

	precision	recall	f1-score	support
anxiety	0.69	0.71	0.70	3260
lonely	0.44	0.37	0.40	1495
normal	0.81	0.86	0.83	3045
stress	0.91	0.90	0.91	1801
accuracy			0.74	9601
macro avg	0.71	0.71	0.71	9601
weighted avg	0.73	0.74	0.73	9601

Random Forest

Accuracy: 76.29%

Classification report:

	precision	recall	f1-score	support
anxiety	0.70	0.74	0.72	3260
lonely	0.43	0.37	0.40	1495
normal	0.87	0.90	0.88	3045
stress	0.94	0.91	0.93	1801
accuracy			0.76	9601
macro avg	0.74	0.73	0.73	9601
weighted avg	0.76	0.76	0.76	9601

Results (ML - tuning)

In the logistic regression: tune the C value, and the solver using Gridsearch method.
Accuracy: 74.89%
Classification report:

	precision	recall	f1-score	support
anxiety	0.72	0.72	0.72	3260
lonely	0.46	0.34	0.39	1495
normal	0.77	0.90	0.83	3045
stress	0.93	0.88	0.90	1801
accuracy			0.75	9601
macro avg	0.72	0.71	0.71	9601
weighted avg	0.74	0.75	0.74	9601

In the linear SVC: tune the C value using Gridsearch method.

Accuracy: 75.29%
Classification report:

	precision	recall	f1-score	support
anxiety	0.72	0.75	0.74	3260
lonely	0.51	0.29	0.37	1495
normal	0.76	0.90	0.82	3045
stress	0.93	0.88	0.91	1801
accuracy			0.75	9601
macro avg	0.73	0.71	0.71	9601
weighted avg	0.74	0.75	0.74	9601

Results (Deep Learning)

- First trial using: 1 LSTM layer with training the model on batch size = 64
  Accuracy: 76.1%, Loss: 0.686
- Second trial using: 2 LSTM layers with training the model on batch size = 64
  Accuracy: 74.4%, Loss: 1.150
- Third trial using: 1 GRU layer with training the model on batch size = 64
  Accuracy: 74.9%, Loss: 1.124
- Fourth trial using: 2 GRU layers with training the model on batch size = 64
  Accuracy: 73.5%, Loss: 1.314
- Fifth trial using: 1 GRU layer, 1 LSTM layer with training the model on batch size = 64
  Accuracy: 74.4%, Loss: 1.331

Findings

Findings

- Data exploration helps in understanding the dataset more.
- Cleaning the text helps in reducing the dimensionality of the features, and helps in getting better performance with the machine learning models and the deep learning models.
- Tuning the hyperparameters helps the models to achieve better results.
- In my case training the model on batch size = 64 gave better results than training the model on batch size = 128

Thank you