

DATA REPORT

CRISP-DM METHODOLOGY

DEVELOPING A PREDICTIVE MODEL FOR EARLY DETECTION OF MENTAL HEALTH CONDITIONS

GROUP 3 MEMBERS:

ELVIS WANJOHI

JESSICA GICHIMU

JESSE NGUGI

STEPHEN GACHINGU

LATIFA RIZIKI

NOVEMBER 2025

TABLE OF CONTENTS

1. Business Understanding.....	1
1.1 Business Overview	1
1.2 Problem Statement.....	1
1.3 Business Objectives	1
1.3.1 Main Objective	1
1.3.2 Specific Objectives	2
1.3.3 Research Questions.....	2
1.4 Success Criteria	3
2. Data Understanding	4
2.1 Data Source	4
2.2 Data Description	4
2.3 Data Quality Checks	4
3. Data Preparation.....	5
3.1 Preprocessing with English text	5
3.1.1 Normalizing and Cleaning.....	5
3.1.2 Tokenization and Stopword Removal.....	6
3.1.3 Removing Punctuation	6
3.1.4 POS Tagging and Lemmatization	7
3.2 Plotting the Distributions	7
4. Modeling.....	9
4.1 Multi-Class Classification with Traditional Models	9
4.2 Model Training.....	10
4.2.1 Models Used.....	10
4.2.2 Hyperparameter Tuning	11
4.2.2 Model Training Pipeline.....	11
4.3 Deep Learning Models	12
4.3.1 Overview.....	12
4.3.2 Data Preparation.....	13
4.3.3 Modeling Setup.....	13
4.3.4 Training and Validation	14
4.3.5 Final Model Evaluation and Saving	14

5. Evaluation	16
5.1 Evaluation Results	16
5.2 Evaluation Methodology	16
5.3 Next Steps.....	17
6. Deployment.....	18
6.1 Deployment Strategy and Hosting.....	18
6.2 Application Workflow and User Functionality	18
6.3 Deployment Challenges and Resolutions	19
6.4 Monitoring and Maintenance	19
6.5 Ethical and Responsible Use Considerations	20
7. Conclusion and Recommendations.....	21
7.1 Conclusion.....	21
7.2 Recommendations.....	21

1. Business Understanding

1.1 Business Overview

Mental health conditions are widespread, yet early signs often appear in everyday language before individuals seek clinical help. Social media posts provide real-time signals of emotional distress that can support timely intervention. This project classifies English and Swahili posts into seven mental health categories; anxiety, bipolar, stress, personality disorder, normal, depression and suicidal to help identify risk patterns that can inform awareness, screening and support strategies.

1.2 Problem Statement

Mental health expressions often appear in everyday language long before individuals seek professional help, but there is no scalable way to analyze this text and identify risk early. Current assessments rely heavily on manual evaluation, which can be slow, subjective and limited to small samples.

This project builds a text classification model that categorizes posts into seven mental health labels: anxiety, bipolar, stress, personality disorder, normal, depression and suicidal. The goal is to provide a structured, automated layer of analysis that supports early screening, improves triage and strengthens outreach efforts. The model is not a replacement for clinical diagnosis but a decision-support tool that can guide professionals and increase public awareness where mental health risks are present.

1.3 Business Objectives

1.3.1 Main Objective

The main objective is to develop a machine learning model that can accurately classify mental health conditions based on textual statements expressed by individuals.

1.3.2 Specific Objectives

To achieve the main objective, the project has the following specific objectives:

1. Translate all text data into Swahili to localize the dataset and improve inclusivity.
2. Identify the most common mental health condition.
3. Preprocess the data through processes such as Vectorization and tokenization, handling missing values, and creating new features such as characters, words and sentences.
4. Use exploratory tools such as word clouds to visualize commonly terms associated with specific mental health categories.
5. Analyze text length to classify a mental health condition or show correlation with a mental health condition.
6. Evaluate model performance using metrics such as Precision, Recall, F1score, Accuracy Score and ROC-AUC.
7. Compare different classification models to determine which performs best for this dataset.
8. Scrapping data from an online platform like twitter to show the efficiency of the model.
9. Create a translate feature to allow English–Swahili switching for interpretability and diversity in the model.

These objectives guided the modeling and evaluation stages of the CRISP-DM process.

1.3.3 Research Questions

To ensure the analysis directly addresses the business problem, the following research questions were defined:

1. Can the dataset be effectively translated and localized to Swahili?
2. Which is the most common health condition?

3. Which features influence mental health conditions?
4. Which words are specific to each mental health category?
5. Which classifier model achieves the best Precision, Recall, F1 score, Accuracy and ROC-AUC?
6. Which classification model performs best for this dataset?
7. How efficiently can the model classify conditions when applied to Twitter data?
8. How can we ensure diversity, fairness and interpretability in the multilingual model?

The project answers these research questions through data exploration, Exploratory Data Analysis (EDA), data cleaning, Swahili localization, feature engineering and vectorization, model training and evaluation, and model interpretation. The resulting insights and classifiers support early screening, triage and multilingual decision-support for mental health awareness and outreach.

1.4 Success Criteria

Project success will be measured by both model performance and practical impact. The model should accurately classify text into the correct mental health category so risks can be identified early and support decisions can be made in time. From an application perspective, success means clear insights into the most common conditions, a high-performing multilingual model and an output that remains interpretable, culturally sensitive and ethically responsible in both English and Swahili contexts.

2. Data Understanding

2.1 Data Source

This project uses the English Mental Health Text dataset from Hugging Face to build a mental health condition classification model. The dataset was translated into Swahili to show the disparity between English and Swahili text. The dataset includes 103,488 rows and 3 columns, which are all of object data type.

2.2 Data Description

The dataset contains translated user statements intended for multilingual mental health text classification. Each row represents one post and includes three columns:

- **english_text:** The original English statement describing a mental health condition or experience.
- **mental_health_label:** The labeled mental health category or condition.
- **swahili_text:** The Swahili translation of the English statement, making the dataset more inclusive and relevant for mental health analysis within the African context.

These fields provide the foundation for analyzing linguistic patterns across mental health categories and for developing an effective multilingual classification model.

2.3 Data Quality Checks

The dataset was assessed for quality before modeling.

- Checked for missing values in the dataset.
- Checked for duplicate rows and inconsistency in the dataset.
- Checked the data type of each column in the dataset.
- Checked for uniformity of data in the dataset.

3. Data Preparation

3.1 Preprocessing with English text

This section ensures the text is structured, consistent and ready for feature extraction and modeling, while retaining the Swahili translation for interpretability and future model expansion. The steps include:

- Normalization: Involves converting the text data into a consistent format by converting all the text data to lowercase.
- Cleaning: Removes URLs, @mentions and hashtags if any appear in the dataset. It expands contractions, normalizes repeated letters and strips special characters. In addition, it standardizes punctuation and whitespace.
- Tokenization: This involves breaking the texts into smaller words or phrases that the model can understand.
- Stopword Removal: This involves removing words with no significant meaning.
- POS Tagging: It is short for Part of Speech and involves assigning each text to a grammatical category like Noun, Verb and Adjective.
- Lemmatization: It reduces words to their base root or form.

3.1.1 Normalizing and Cleaning

- Dropped interim helper columns: english_length, swahili_length, english_word_count, english_sentence_count, english_char_count, swahili_word_count, swahili_sentence_count, swahili_char_count.
- Retained core fields: english_text, swahili_text, mental_health_label.

- A new column named `cleaned_text` was created from the `english_text` column, where text was converted to lowercase and stripped of leading and trailing whitespace to ensure consistent formatting.
- Noise such as URLs, special characters and unnecessary symbols was removed using a regex-based cleaning step. This resulted in cleaner input for downstream preprocessing.
- Preview checks confirmed that the dataset now contains standardized text ready for further feature engineering and model training.

3.1.2 Tokenization and Stopword Removal

In this step, the text is broken down into individual tokens and cleaned by removing words that do not contribute meaningful information to the model.

- A `RegexpTokenizer` was applied to split the text using a regular expression, ensuring that meaningful contractions such as “I’m” and “don’t” remain intact.
- A custom function was created to remove English stopwords from the tokenized text.
- Single character tokens and isolated short words were filtered out to reduce noise.
- The cleaned tokens were joined back into a structured text format. This preserved only the words relevant for mental health classification.

This step helped reduce vocabulary size, improve text quality and prepare the data for vectorization.

3.1.3 Removing Punctuation

- After tokenization and stopword removal, punctuation was removed from the text to ensure that all remaining tokens were processed consistently.

- A regular expression was applied to strip characters such as commas, apostrophes and symbols while retaining meaningful words for analysis.

This step helped eliminate remaining punctuation attached to stopwords and ensured a cleaner and standardized input for vectorization and model training.

3.1.4 POS Tagging and Lemmatization

- Part-of-Speech (POS) tagging was applied to assign grammatical roles to each word before lemmatization. This step ensures that words are reduced to their correct base form according to context. An example is running to run when tagged as a verb.
- WordNet's lemmatizer was used together with POS tagging to convert words into their root forms. This improved consistency and reduced vocabulary sparsity in the model.

This step improves the model's accuracy by making the text more consistent while still keeping the original meaning of the words related to mental health.

3.2 Plotting the Distributions

Exploratory Data Analysis was performed by visualizing the relationships between the variables. Some of the key visualizations done include:

- **Mental Health Label Distribution Analysis:** This plot shows how the mental health categories are distributed across the dataset. Anxiety is the most common condition with 17,620 posts, followed by normal with 16,067 posts, depression with 15,900 posts and stress with 15,229 posts. The least represented categories are personality disorder with 13,912 posts, bipolar with 13,708 posts and suicidal with 11,045 posts. This shows a clear class imbalance that will need to be addressed before model training.

- **Text Length Distribution Analysis:** This plot illustrates the distribution of text lengths across both English and Swahili posts. Most entries in both languages contain fewer than 50 words, showing that the majority of posts are short and concise. English posts have a wider spread with a higher average word count of about 82 words, while Swahili posts cluster more tightly around a lower range of about 29 words. The log scale highlights a long-tail pattern caused by a small number of posts extending into the hundreds or thousands of words. Overall, the distribution shows that most mental health related posts are brief with only a few extended messages.
- **Text Length by Mental Health Label:** This plot compares the average number of words used in posts across the different mental health labels. Depression posts have the highest average word count at about 175 words per post, followed by suicidal posts at about 152 words. Anxiety and bipolar posts have moderate lengths of about 72 words and 67 words respectively, while personality disorder and stress posts are shorter with about 53 and 50 words on average. Normal posts are the shortest, averaging only about 30 words. This shows that users expressing depressive or suicidal thoughts tend to write longer and more descriptive messages while posts about stress or normal states are brief and concise.

4. Modeling

Several text-classification models were evaluated for multi-class mental health prediction. These models represent a progression from interpretable linear baselines to nonlinear ensembles. This allows for comparison across scalability, generalization and performance on sparse text features:

- **Logistic Regression:** Baseline linear model that is interpretable and provides usable probability estimates for decision making.
- **Multinomial Naive Bayes:** Fast probabilistic baseline well suited to word-frequency features and efficient to train.
- **Linear Support Vector Machine (LinearSVC):** Large margin linear classifier that performs strongly on high-dimensional TF-IDF text.
- **Random Forest:** Nonlinear tree ensemble included to test whether feature interactions add value in the multi-class setting.

In addition to these traditional machine learning models, a transformer-based model (RoBERTa) is trained in a separate deep learning notebook to establish an advanced benchmark using contextual embeddings.

4.1 Multi-Class Classification with Traditional Models

- Before training the traditional machine learning models, the dataset was checked for duplicated text entries to ensure that no repeated records would bias model learning.
- Duplicate rows were identified using the cleaned_text column and subsequently removed, leaving 99,840 unique observations to be used for modeling.
- A further check confirmed that no missing values were present in either the feature or target columns, meaning no imputation was required at this stage.

- The target variable which is the `mental_health_label` was then encoded into numerical form using `LabelEncoder`. This resulted in seven integer classes ranging from 0 to 6.
- A class distribution assessment showed that the dataset is relatively balanced with each class containing more than 10,000 samples. This allowed the models to be trained without applying any resampling techniques.
- To prepare the data for training, the feature matrix `X` was defined as the cleaned text and the target vector `y` as the encoded labels.
- The dataset was then split into training, validation and test sets using a 70% / 15% / 15% stratified split, ensuring that the label proportions remained consistent across all subsets.

This split structure allows the training set to be used for fitting the models, the validation set for hyperparameter tuning and model selection, and the test set for final performance evaluation. The use of stratification and a fixed random state ensures that the split is both reproducible and free from class imbalance drift.

4.2 Model Training

4.2.1 Models Used

The following four traditional machine learning classifiers were selected for multi-class mental health text classification: Logistic Regression, Multinomial Naive Bayes, Linear Support Vector Classifier (`LinearSVC`) and Random Forest.

Each model was first trained using a baseline configuration, after which hyperparameter tuning was conducted to identify the best performing version of each model.

4.2.2 Hyperparameter Tuning

Each classifier was tuned using a focused search to assess how key settings influence performance while controlling training cost.

- **Logistic Regression:** The search varied the regularization strength, applied the standard L2 penalty, enabled balanced class weights to treat all classes fairly and used a solver suitable for large datasets.
- **Multinomial Naive Bayes:** The search adjusted the smoothing level to regulate the influence of rare words.
- **Linear SVC:** The search varied the regularization strength, compared the hinge and squared hinge loss functions, and applied balanced class weights.
- **Random Forest:** The search explored model capacity by changing the number of trees from 200 to 300 and the maximum tree depth from 15 to 25.

Across all pipelines, the TF-IDF vocabulary size was tuned between 80,000 and 90,000 terms to compare narrower versus wider text representations while keeping leakage controls and evaluation settings consistent.

4.2.2 Model Training Pipeline

A modelling pipeline was implemented to streamline preprocessing, hyperparameter tuning and performance evaluation for all traditional machine learning classifiers.

- The pipeline first applies the TF-IDF vectorizer which converts the cleaned text into numerical features using a unigram bigram range to preserve both individual words and short contextual phrases. This transformation produces a sparse matrix that is suitable for model training at scale.

- The transformed data is then passed into each classifier through a GridSearchCV wrapper. This enables systematic hyperparameter tuning and cross-validated performance comparison under identical conditions.
- For every model, the best performing configuration is extracted, saved using joblib for reproducibility and used to generate predictions on the training, validation and test splits.
- The pipeline also records multiple evaluation metrics including accuracy, macro-averaged F1-score, precision, recall and AUC where supported. This ensured that both overall performance and class-level balance are shown.

By combining text vectorization, model training, hyperparameter tuning, prediction and metric logging into one automated workflow, the pipeline keeps the process consistent, reduces manual errors and allows a fair comparison of all models before moving to model evaluation.

4.3 Deep Learning Models

4.3.1 Overview

This section fine tunes a RoBERTa transformer for seven class mental health text classification. RoBERTa, an enhanced variant of BERT, learns context aware language representations and performs reliably on large datasets.

- The workflow loads and cleans the data, encodes the target labels into numbers and applies a stratified split into training, validation and test subsets to preserve class balance.
- The RoBERTa tokenizer then converts the cleaned text into tokens. This allows the model to process each sentence in context.

- A sequence classification head is added to RoBERTa and training is managed by a Trainer that handles batching, optimization, scheduled evaluation on the validation split, logging and checkpointing.
- Accuracy is tracked throughout training and learning curves for training and validation accuracy and loss are generated to monitor convergence and detect overfitting. The best model is saved for reproducibility and for later evaluation.

4.3.2 Data Preparation

- The dataset was first loaded and reviewed to confirm that the cleaned text column contained no missing values and that each record was paired with a valid mental health label.
- The target labels were then converted into numeric form to support supervised learning. A stratified split was applied to create training, validation and test sets while preserving the original class distribution across all seven categories.

This structure ensures that the model learns from representative data, is tuned on unseen data and is finally evaluated on a fully held out set. This supports fair measurement of generalization performance and prevents information leakage.

4.3.3 Modeling Setup

A RoBERTa transformer was selected as the deep learning model for this task due to its strong performance on text classification and its ability to capture context within sentences.

- The RoBERTa tokenizer was used to convert each cleaned text entry into token IDs and attention masks. This allows the model to process the input in a format it can understand.
- The pretrained RoBERTa base model was then loaded and extended with a classification head configured for seven output classes matching the encoded mental health labels.

- The model was set up to run within the Hugging Face Trainer framework which handles batching, optimization, periodic evaluation and checkpoint management during training.

This setup ensures a consistent workflow, reduces manual configuration and supports reproducible experimentation across multiple runs.

4.3.4 Training and Validation

- The model was trained on the stratified training set while the validation set was used to monitor performance at the end of each epoch.
- The training configuration included the batch size, learning rate, number of epochs and evaluation schedule. This allowed the model to update its parameters gradually while tracking progress.
- During training, both accuracy and loss were recorded for the training and validation splits. Learning curves were generated to visualize whether the model was improving steadily or beginning to overfit.
- The validation metrics provided an early indicator of generalization quality, guiding decisions on whether the training duration or learning rate required adjustment.

This process ensured that the final model selected for testing reflected the best balance between learning capacity and stability rather than simply memorizing the training data.

4.3.5 Final Model Evaluation and Saving

After training, the selected RoBERTa model was evaluated on the held out test set to provide an unbiased measure of its final performance. The test results confirmed that the model was able to generalize beyond the training and validation splits achieving a high level of accuracy on

previously unseen text. This evaluation completes the modeling phase and validates the effectiveness of the deep learning approach for multi-class mental health classification.

To support reproducibility and future use, the trained model and its tokenizer were saved. This allows the model to be reloaded without retraining, enables consistent inference across different environments and prepares for later deployment or comparison with additional models.

5. Evaluation

This section compares all trained models on the held out test set, identifies the top performer and justifies the model selected for deployment. Evaluation follows a fair comparison protocol that includes identical preprocessing and splits, validation used strictly for tuning and no further adjustments after test exposure. Performance is summarized using accuracy, macro F1-score, precision, recall and confusion-matrix insights to show overall and per class behaviour. The set includes Logistic Regression, Multinomial Naive Bayes, Linear SVC, Random Forest and a RoBERTa transformer.

5.1 Evaluation Results

All models were evaluated on the same test set to ensure a fair performance comparison. Among the traditional machine learning models, Linear SVC recorded the strongest results. The model achieved the highest test accuracy and macro F1-score within that group, followed by Logistic Regression, Multinomial Naive Bayes and Random Forest. The traditional models showed clear signs of overfitting.

The RoBERTa transformer outperformed all traditional models achieving a validation accuracy of 88.59% and showing stronger generalization across all seven mental health classes. The confusion-matrix patterns showed that RoBERTa reduced misclassification between categories with similar language patterns, particularly depression, stress and suicidal posts where classical models showed higher overlap.

5.2 Evaluation Methodology

The evaluation followed a controlled and reproducible setup to ensure that model comparisons were unbiased. All models were trained and tested on the same cleaned dataset, using the same

stratified train validation test split to preserve the original class distribution. No resampling or data augmentation was applied after the split which prevented information leakage.

For the traditional models, text was converted into numerical features using TF-IDF while RoBERTa used a tokenizer that preserved word order and context. Each model was tuned only on the validation set and the test set was kept fully unseen until final evaluation. Performance was assessed using accuracy, macro F1-score, precision, recall and confusion-matrix patterns. This allowed both overall performance and class level behaviour to be examined.

This approach aligns with CRISP-DM standards by separating tuning, testing and interpretation stages to maintain reliability in results.

5.3 Next Steps

- **Primary model:** Deploy RoBERTa for seven class mental health classification. It achieves 88.59% validation accuracy and reduces misclassifications in sensitive categories such as depression and suicidal, outperforming the traditional models.
- **Probability use cases:** When probability thresholds are required, apply a simple calibration step to RoBERTa outputs or retain a Logistic Regression variant to provide calibrated scores for decision rules.
- **Monitoring:** Track accuracy and macro-F1 by class on live data. Watch for shifts in class distribution and declines in precision for high risk categories, and trigger review and retraining if macro-F1 drops by three to five points or if drift is detected.
- **Maintenance:** Use the saved RoBERTa model and tokenizer for API deployment. Refresh the training data with recent posts, re-evaluate with the held out protocol and schedule periodic retraining when vocabulary changes, domain drift or performance degradation appear.

6. Deployment

6.1 Deployment Strategy and Hosting

The final model, RoBERTa transformer, was deployed as an interactive Streamlit web application. It allows users to input free text and receive an instant mental health category prediction. Instead of storing the trained model in the GitHub repository, the deployment loads both the model and tokenizer directly from Hugging Face. This approach avoids large binary model files, prevents GitHub size limits and enables easier version control using the hosted model repository. Streamlit Cloud automatically installs all dependencies from requirements.txt and launches the app using app.py.

The live application is accessible at:

<https://globalmentalhealthapp-niqlikscywvz7gh3ryy65hyy.streamlit.app/>

6.2 Application Workflow and User Functionality

The deployed app performs the full end-to-end prediction pipeline in real time:

1. User enters text in any language.
2. Text is auto translated to English using GoogleTranslator.
3. The text is tokenized and passed through the RoBERTa model for prediction.
4. The predicted class is mapped to one of seven mental health categories.
5. The app displays helpful and category specific resources. An example is helplines for suicidal cases.
6. A safety disclaimer reminds users that the tool does not replace professional mental health support.

This structure makes the model usable by any non-technical audience.

6.3 Deployment Challenges and Resolutions

- **GitHub file size limitation:** The trained model could not be uploaded as .pkl or .sav so it was instead hosted on Hugging Face and loaded automatically when the app starts.
- **Streamlit build failure:** The app initially failed on deployment due to incorrect file paths during installation. Updating the dependency list and removing unused local model references resolved the issue.
- **Translation dependency:** Since user text may not be in English, translation was added to avoid misclassification caused by multilingual inputs.

6.4 Monitoring and Maintenance

The deployed model will be monitored for both technical reliability and classification quality. Key practices include:

- Track prediction performance over time, focusing on macro F1-score and class level accuracy. This is especially for high risk classes like depression and suicidal.
- Review user inputs periodically to detect language drift, emerging slang or new mental health phrasing not present in the training data.
- Retrain or fine tune the model if performance drops, if class balance shifts or if new data significantly changes expression patterns.
- Maintain pinned library versions in requirements.txt to prevent dependency breakage during rebuilds.
- Update the Hugging Face model repository rather than redeploying the app, keeping Streamlit lightweight and stable.

6.5 Ethical and Responsible Use Considerations

Due to the model dealing with sensitive mental health content, the deployment includes safeguards to prevent misuse and misinterpretation. The application displays a clear disclaimer stating that predictions are not medical advice and should not replace professional diagnosis or emergency support. For high risk outputs such as suicidal, the interface provides crisis support contacts and encourages immediate help seeking rather than relying on automated decisions.

No user text is stored, logged or reused for training, preserving privacy and preventing unintended data collection. Any future expansion of the system will follow responsible AI principles. This includes bias evaluation, transparent model updates and protection of user input.

7. Conclusion and Recommendations

7.1 Conclusion

This project followed the CRISP-DM framework to build and evaluate a multi-class mental health text classifier capable of identifying seven conditions from user generated language. A RoBERTa transformer emerged as the best performing model. It achieved the highest test accuracy and strongest class level balance particularly in high risk categories.

The project met its objectives by delivering a scalable and multilingual classification system. This showed clear performance gains over traditional machine learning models. The insights can support early screening, mental health awareness and decision support in real world applications.

7.2 Recommendations

1. Refresh the training dataset periodically with new real world text to capture evolving language patterns, slang and mental health expressions not present.
2. Expand multilingual capability beyond English and Swahili to increase inclusivity and improve applicability across additional African contexts.
3. Conduct periodic bias and fairness checks to ensure that the system does not disproportionately misclassify specific groups, dialects or linguistic styles.

These actionable recommendations support safe deployment, long-term model reliability and responsible use of AI for mental health screening and awareness.