Data Report - CRISP DM Methodology

PROJECT TITLE: AFYAMIND EMOTION DETECTION MODEL

1. Business Understanding

The AfyaMind project focuses on developing an automated emotion detection model capable of classifying text-based data into distinct emotion categories. The system is designed to support mental health professionals, researchers, and digital wellness organizations in identifying emotional states expressed in written communication. Understanding the underlying emotions in user text can inform early intervention strategies, improve empathetic response systems, and enable large-scale emotional trend analysis.

Objectives:

Main objective: To build and evaluate a machine learning model that can automatically classify text into emotional categories using the GoEmotions dataset.

Specific Objectives:

- To explore and understand the distribution of emotions in the GoEmotions dataset.
- To transform textual data into numerical form using techniques such as TF-IDF, or contextual embeddings (e.g., RoBERT).
- To build and compare baseline models (e.g., Logistic Regression, Random Forest, Naive Bayes) with deep learning models (e.g., RoBERT).
- To evaluate models using metrics appropriate for multi-label classification (e.g., F1-score, Precision, Recall)
- To deploy the model on Streamlit and integrate it with an existing model to provide intervention.

The project’s success will be evaluated based on the performance metrics, particularly macro F1-score and recall, to ensure balanced prediction accuracy across both majority and minority emotion classes.

2. Data Understanding

The dataset used for the AfyaMind project is publicly available via HuggingFace and includes text, emotion labels and metadata such as comment ID, subreddit, author and timestamp. The dataset comprises 58,009 Reddit comments annotated with 27 emotion categories and Neutral, sourced from Reddit. The emotions range from joy, sadness and anger to surprise, fear and love.

The dataset was stored in CSV format and read into a panda DataFrame for analysis.

An initial review revealed the following structure:

- **Columns**: ‘text’, ‘emotion’, and additional metadata columns depending on the source (such as ID, length, sentiment score).
- **Rows**: Each record corresponds to a single labeled text instance.
- **Label distribution:** Emotions like ‘joy’ and ‘sadness’ were most frequent, while ‘fear’ and ‘surprise’ appeared less often.
- **Data volume**: The dataset contained several thousand entries, sufficient for both classical and deep learning models.

Exploratory data analysis (EDA) identified common challenges typical of social text data:

- Variations in spelling, abbreviations, and slang expressions.
- Presence of emojis and symbols.
- Texts of varying lengths, ranging from single-word responses to multi-sentence messages.
- Imbalanced representation across emotion categories.

Data quality verification steps ensured completeness and consistency:

- Checked for null or empty values in key columns (‘text’, ‘emotion’).
- Removed duplicate records.

- Verified that emotion labels matched predefined categories.
- Conducted sanity checks by sampling text to ensure labeling accuracy and coherence.

3. Data Preparation

Data preparation involved a systematic transformation of raw textual inputs into model-ready formats. This phase was critical for ensuring data consistency, improving model performance, and minimizing preprocessing errors.

The major steps taken were:

Text Cleaning:

All text was converted to lowercase to ensure uniformity. Special characters, digits, URLs, and punctuation were removed using regular expressions. Contractions were expanded and excessive whitespace was eliminated.

Tokenization:

Each sentence was split into tokens using NLTK's word tokenizer for classical models and Hugging Face's tokenizer for transformer-based models. Tokenization preserved word boundaries, enabling downstream models to interpret text sequences effectively.

Stopword Removal:

Common English stopwords such as 'and', 'the', and 'is' were removed to reduce dimensionality.

Lemmatization:

Words were reduced to their base or dictionary form using WordNetLemmatizer. This step helped unify inflected forms, improving consistency in vectorization.

Feature Extraction:

Two parallel representations were prepared:

- TF-IDF Vectorization for classical models, converting text into numerical feature vectors weighted by term frequency and inverse document frequency.
- Token Embeddings for transformer models, creating token IDs and attention masks for use with pre-trained architectures such as RoBERTa.

Handling Missing Data:

Text entries with missing labels were dropped. For any auxiliary columns with missing values, median or mode imputation was applied.

Handling Class Imbalance:

Class imbalance was handled by applying weighted loss functions for transformer models. These techniques ensured that minority classes contributed proportionally to the loss calculation.

Train-Test Split:

The dataset was split into training (80%) and testing (20%) sets using stratified sampling to maintain proportional emotion representation.

Vectorization Verification:

Post-processing checks confirmed that the feature matrix dimensions aligned with the dataset size, ensuring readiness for modeling.

4. Modeling

1. Baselines: Classical Machine Learning Models

Classical models were used as a necessary first step to establish a performance benchmark. They rely on feature engineering (like TF-IDF) to convert text into numerical vectors before training the classifier.

The models we explored are:

- **Logistic Regression:**

Served as the baseline model. This model provided a benchmark for interpretability and speed. It was wrapped in a **MultiOutputClassifier** to handle the simultaneous prediction of the 7 emotion labels.

- **Linear Support Vector Classifier (Linear SVC)**

Finds the optimal hyperplane that maximally separates the data points of different classes in the feature space. Support Vector Machine (SVM): Applied with a linear kernel and adjusted class weights to improve recall for minority emotion classes.

- **Multinomial Naive Bayes (MNB):**

A probabilistic classifier that is well-suited for text classification using word frequencies (like TF-IDF). It's "naive" because it assumes that all features (words) are conditionally independent of each other, given the class.

Because this is a **multi-label** classification problem (a text can have multiple emotions), the MultinomialNB model is wrapped in a **MultiOutputClassifier**. This wrapper strategy trains a separate and independent MNB classifier for each of the seven target emotion labels.

Best Baseline Performer: Linear SVC was trained with **class weighting** to handle the severe imbalance in the emotion classes. It achieved the best F1 score among all classical models (Micro F1: **0.5532**), setting the minimum acceptable performance target for the final model.

2. Deep learning Model

RoBERTa (Robustly Optimized BERT Pretraining Approach) is a transformer-based model. Unlike TF-IDF, which only counts word occurrences, RoBERTa reads the entire sentence and understands the *context* of each word. It's pre-trained on a massive amount of text, so it already has a deep understanding of language. This project uses a fine-tuned 'roberta-base' version.

Data Preparation:

The text is converted into numbers using the RoBERTa tokenizer. This tokenizer separates words into specific tokens, adds special tokens (like [CLS] and [SEP]), and creates an "attention mask" so the model knows which tokens are real words.

A custom 'CustomDataset' is used to prepare the tokenized text and the 7 emotions for training in batches.

Training(MultiLabel Approach):

The model is trained using BCEWithLogitsLoss (Binary Cross-Entropy With Logits Loss). This loss function is the key to the multi-label setup. Instead of forcing the model to pick just one emotion, BCE loss treats each of the 7 emotion outputs as a separate "yes/no" prediction. This allows the model to predict multiple emotions in one text.

The model's weights are updated using the Adamw optimizer which is standard for most transformer models.

Final Model: The model was fine-tuned on our 7-label emotional dataset. It achieved the project's highest overall performance (Micro F1: **0.5930** and lowest Hamming Loss: **0.0834**), making it the chosen model for deployment due to its superior contextual understanding and ability to capture subtle emotional nuance.

5. Evaluation

Performance of Baseline Models

We first established a baseline using traditional machine learning models, primarily focusing on **Linear SVC** and **Logistic Regression**, both wrapped in a MultiOutput architecture to handle the seven Ekman emotions simultaneously

We used the following different metrics to perform evaluation: Accuracy, Precision, Recall, and F1-score. Given the dataset’s class imbalance, macro-averaged F1-score was the primary metric. The confusion matrix provided insights into class-specific errors.

The evaluation findings include:

- ### Logistic Regression:

The model achieved a Micro F1 score of **0.5487**, demonstrating a reasonable overall level of accuracy, while its Macro F1 score was slightly higher at **0.5551**, which indicated a slightly more balanced performance across all seven classes, including the rarer ones. Its error rate, measured by Hamming Loss, stood at **0.1097**.

- ### Linear SVC

The model proved to be the strongest classical baseline, scoring a Micro F1 of 0.5532. This slight edge over Logistic Regression, combined with a Hamming Loss of 0.1065, established the minimum performance standard the final deep learning model had to surpass.

RoBERTA Deep Learning Model

The final production model utilized **RoBERTA (Robustly Optimized BERT Pretraining Approach).** RoBERTA's ability to understand the deep, bidirectional context of language allows it to capture subtle emotional nuances that simpler models miss, leading to a substantial gain in overall effectiveness.

The RoBERTA model recorded the **highest overall accuracy** with a Micro F1 score of **0.5930**. This figure confirms its superior reliability when processing the vast majority of user texts, ensuring that common emotions are flagged with the greatest precision and recall.

Crucially, the RoBERTA model also achieved the **lowest error rate** with a Hamming Loss of **0.0834**. This means that, on average, less than 8.5% of the total labels predicted were incorrect, which is a key measure of system robustness for a client-facing application like AfyaMind.

Further diagnostic evaluations:

We identified specific confusion patterns:

- Misclassification commonly occurred between ‘sadness’ and ‘fear’ due to overlapping linguistic expressions.
- Short texts with ambiguous sentiment were occasionally misclassified as ‘neutral’ when no explicit emotion cue was present.
- The transformer model reduced such ambiguity by leveraging contextual embeddings that capture subtle word relationships.

Hyperparameter tuning results indicated diminishing returns beyond five epochs for transformers. Early stopping improved generalization and prevented overfitting.

Overall, the transformer-based model met the success threshold defined during Business Understanding and demonstrated readiness for deployment.

6. Deployment

The saved RoBERTA model was deployed using Streamlit, which is an open-source Python framework that simplifies the sharing of custom web applications for machine learning projects

The deployed Streamlit application, "AfyaMind," is a multi-page tool with a custom CSS theme designed to support mental wellness. It provides users with four main utilities: "Single Text Analysis," "Batch Analysis" (via text or CSV upload), a "Mental Health Chatbot," and a "CSV Intervention Generator."

The Batch Analysis via CSV can be utilized by brand analysts to analyze the sentiment towards their brand or company.

Beyond simple classification, the application's primary function is to integrate the RoBERTa model's output with a generative AI. The detected emotion labels (e.g., "anger," "sadness") are not the final result but are instead used to programmatically construct a detailed prompt for the DeepSeek API (deepseek-chat). This prompt instructs the large language model to act as a compassionate, Kenyan-focused mental health assistant and provide supportive, actionable advice based on Cognitive Behavioral Therapy (CBT) principles.

This two-step system (classification by RoBERTa, and intervention by DeepSeek) is made robust by a critical fallback mechanism. If the generative API fails or is not configured, the script defaults to providing static, rule-based supportive messages and, most importantly, directs the user to a list of official Kenyan mental health hotlines and resources. This is a prime example of how the emotion detection model can be integrated with an existing model with developers.