Minimum Viable Product (MVP) Description for Data Science & Engineering - Version 0.1

Team Name: Group 1

Date: 2023/08/16

---

1. MVP Objective :

The primary goal of our MVP is that the chatbot should accurately understand customer queries in all three languages. This involves training the model to classify the intent of the query, extract relevant entities, and accurately determine the language being used. The MVP should also generate relevant responses in the corresponding language. The response generation process involves taking the input query, understanding its intent and context, and generating a response that provides the required information.The MVP should be capable of handling and processing multilingual data seamlessly. This includes preprocessing text data in different languages, handling language-specific features, and ensuring the model's ability to switch between languages based on the customer's input.

---

2. Data Understanding & Preliminaries:

- **Data Sources**: The primary dataset for our intelligent trilingual chatbot project would consist of customer interactions between bank customers and support agents in Sinhala, Tamil, and English languages.There are many banks that support all three languages on their websites. Furthermore there are many FAQ sections (some need to be translated) which can be directly applied into our use.The data can be utilized using a web scrape. Bank leaflets also come in different languages and have information that directly relates to informing customers about various services.Some more data regarding bank procedure and services can be obtained by contacting a bank. We require a good idea of bank procedures to accurately identify steps in user stories.

- **Data Challenges**: There are potential challenges that need to be addressed when working with such a diverse and multilingual dataset such as,
    - The quality of translations can impact the accuracy of results.
    - There might be class imbalances where certain types of queries are more common than others.
    - Some interactions might have missing data fields or incomplete sentences, which can affect the quality of training data and subsequent model performance.
    - Unusual or outlier interactions might be present in the dataset.

    - Capturing and understanding the context of conversations can be challenging.
    - Some customer queries might be unclear, misspelled, or use informal language.

---

3. Key Data Processing & Feature Engineering Steps:

- Step 1: Data Preprocessing - Prepare the collected data by performing tasks such as text normalization, tokenization, and removing noise.
- Step 2: Language Identification and Segregation - Split the dataset into separate language-specific subsets.
- Step 3: Translation - Split the dataset into separate language-specific subsets.
- Step 4: Intent Classification and Entity Extraction - Identify the intent of customer queries and extract relevant entities
- Step 5: Response Generation - Generate appropriate responses based on the identified intent and extracted entities.
- Step 6: Feature Encoding (Language) - Convert textual data into numerical representations that can be used by machine learning models.
- Step 7: Model Training and Validation - Train machine learning or neural network models on the preprocessed data.
- Step 8: Evaluation and Iteration - Evaluate the chatbot's performance and make iterative improvements.

---

4. Model/Algorithm Selection:

- **Model/Algorithm Chosen**:Rasa framework, an open-source NLP framework, will be used to develop the chatbot's language understanding and dialogue management capabilities. Pre-trained language models using neural networks, such as those based on LSTM (Long Short-Term Memory) will be used too.
- **Rationale**: Rasa framework was picked to develop the chatbot due to its open-source nature, flexibility, multilingual support and customizabilty.
- **Evaluation Metric**: Intent Classification will be evaluated using Accuracy, Precision, Recall and F1-score. Response Generation will be evaluated using user feedback, BLEU Score and Contextual Understanding.

---

5. Expected Outcomes & Visualizations:

- The intent classification performance of the chatbot's model will be assessed using metrics such as accuracy, precision, recall, and F1-score. These metrics will provide insights into how well the model is able to correctly classify the intent of customer queries for each language

- A confusion matrix will be created for each language to visually represent the classification results. Each row in the matrix corresponds to the actual intent category, while each column corresponds to the predicted intent category.

---

6. Assumptions & Constraints:

- **Assumptions:**
  - Assuming that the translations are accurate and contextually meaningful.
  - Assuming that the collected dataset of customer interactions covers a diverse range of query types, intents, and language variations.
  - Assuming that the intent labels assigned to customer queries during data preprocessing are accurate.
- **Constraints:**
  - Security of user data and transactions must always be guaranteed.
  - The chatbot needs a thorough understanding of banking terms, products, and services.
  - The chatbot might not understand and respond to complex emotional interactions.

---

7. User Interaction & Deployment :

- **Usage Scenario:** The user can initiate a conversation with the chatbot by typing a query in Sinhala, Tamil, or English. The chatbot will analyze the input, classify the user's intent, extract any necessary information, and generate a relevant response in the same language.
- **Deployment Strategy:**
  - Create user interfaces for both web and Telegram bot platforms.
  - Develop an API endpoint that accepts user queries as input and returns chatbot responses. This endpoint will connect the front-end interface with the model.
  - When a user submits a query, the front-end interface sends a request to the API endpoint with the user's input text.
  - The API first identifies the language of the input text to determine whether it's in Sinhala, Tamil, or English.
  - The API then processes the input using the trained model. It classifies the intent and generates an appropriate response in the same language.
  - Generated response is sent back to the front-end interface, where it's displayed to the user as a chatbot reply.

---

8. Future Iterations & Scalability:

- Collect user feedback and use it to fine-tune the model's responses.

- Continuously update the model with new user interactions to improve its accuracy and relevance over time.
- Implement advanced preprocessing techniques to handle noisy text, including misspellings, informal language, and abbreviations commonly used in customer queries.
- Analyze patterns in user interactions to identify frequently asked questions, emerging trends, and areas where the chatbot can provide added value.