1. **Collect Data:** Collecting a dataset that includes examples of user inputs and the corresponding services they needed. Use the template made by Greg as base and expand on this.
2. **Preprocess Data:** Clean text data by removing stop words, punctuation or other additional string. Convert the text into a format suitable for machine learning models.
3. **Feature Extraction:** Use techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings like Word2Vec or GloVe to represent text data as numerical features.
4. **Selecting a Model:** Try one of the following methods:
   * **Naive Bayes:** Simple and efficient for text classification tasks.
   * **Support Vector Machines (SVM):** Effective for text classification with non-linear decision boundaries.
   * **Deep Learning Models (e.g., LSTM, GRU, or Transformer):** Powerful for capturing complex patterns in sequential data.
5. **Train the Model:** Split dataset into training and testing sets. Train chosen model on the training data and evaluate its performance on the testing data. Fine-tune hyperparameters if necessary.
6. **Integration with WOF App:** Once the model is trained and performs well, integrate it into the app using fastAPI. When a user inputs an issue from flutter, use the trained model to predict the service needed based on their input.

Sample code skeleton using scikit-learn and a Naive Bayes classifier:

python

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, classification\_report

# Step 2: Data Collection and Preprocessing

# ... Load and preprocess your dataset ...

# Step 3: Feature Extraction

vectorizer = TfidfVectorizer()

X = vectorizer.fit\_transform(text\_data)

y = labels

# Step 4: Choose a Model and Train

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = MultinomialNB()

model.fit(X\_train, y\_train)

# Step 5: Evaluate the Model

y\_pred = model.predict(X\_test)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

# Step 6: Integration with WOF App