**Item Categorization**

### **Problem Statement:**

Create a model or research the necessary steps to create a model for

categorizing items. When the cook adds an item to their kitchen, it should be automatically

categorized into multiple categories. We can provide the sample data for this to train the model.

For instance:

• Idly - South Indian, Protein Rich, Breakfast, Baked Items etc.

• Chicken Vindaloo - North India, Punjabi, Non-Veg, Chicken, Protein Rich etc.

• Ragi Dosa - South Indian, Diabetic Friendly, Millet Based, Pregnancy friendly

Etc.

### **Approach:**

* Create/Find dataset
* Use the BERT model to perform Multi-label classification
* Create a frontend for interacting with the user
* Connect the frontend and the backend

### **Implementation details:**

* The first step to create a model is to find a Dataset that our model can be trained on. We created a dataset containing dish names along with binary indicators for various categories ensuring that the dataset is balanced and representative of different cuisines and dish types.
* Utilize the BERT (Bidirectional Encoder Representations from Transformers) model for text classification as it offers state-of-the-art performance in natural language processing tasks.
* Fine-tune a pre-trained BERT model on the dish names dataset to perform multi-label classification into the specified categories.
* Trained the BERT-based model on the dataset using PyTorch, ensuring to validate the models performance using the Training loss and Validation loss of the models per epoch.
* Saved the trained BERT model using Python's pickle.
* This step ensures that the trained model can be saved and loaded efficiently for future use.
* Designed a user-friendly interface using HTML, CSS, and JavaScript (potentially with frameworks like React or Vue.js) to provide a seamless user experience.
* Integrate the Python code, including the BERT-based model, into the backend of the UI.
* Establish communication between the frontend and backend components to send user input (dish names) to the Python code and display the output (predicted categories) back to the user on the UI.

### **Problems faced:**

* Since we were unable to find a proper dataset, we had to create a dataset that would be appropriate to the problem statement. We did this by combining the dishes from various food datasets as well as populating their protein richness etc.
* Creating a multi-label classification model was a new concept that we did not deal with before so we had to find a model that would help us with that and understand its working.
* Combining the pickle file of the model to the python code to predict the users input.

### **Future Improvements:**

### Enhance dataset diversity.

* Deploy the model on cloud platforms for scalability and accessibility.
* Better UI Integration.
* Combine predictions from multiple models for improved accuracy.