

Problem Statement:

People find it difficult to get the right fitness advice from individuals around them. Hence there is a necessity for a source to provide the right fitness advice catering to the needs of the individual.

Libraries involved:

Hugging Face Transformers (transformers):

GPT2LMHeadModel: This is a class from the Hugging Face Transformers library that represents a pre-trained GPT-2 model for language modeling.

GPT2Tokenizer: It is used for tokenizing and encoding text for use with the GPT-2 model.

GPT2Config: It provides the model configuration for the GPT-2 model.

TextDataset and DataCollatorForLanguageModeling:

TextDataset: A class from Transformers for processing text data as a dataset.

DataCollatorForLanguageModeling: It prepares data for language modeling by batching and tokenizing text data.

Trainer and TrainingArguments:

Trainer: This class from Transformers is used for training machine learning models, such as the GPT-2 model.

TrainingArguments: It contains training-specific arguments and options for the Trainer.

torch (PyTorch): This is the deep learning library used for creating and training neural network models.

pickle: It is a Python library used for serializing and deserializing Python objects, which is used in this code to save the fine-tuned model to a file.

Interaction:

Once the web app is set up, the user sends the message as an API request from the frontend using Axios to the flask app which is running.

The flask app handles the POST and GET requests between the AI model and the application.

The sent message from the user is given to the FitnessBot model and the respective response is obtained.

This response is now passed through the flask app to the frontend react website and thus the output is displayed.