**Title: Task Assignment for AI Generative Role**

**Objective:**

This task evaluates the candidate's ability to develop an AI-driven chat model that operates directly in the terminal. As a fresher, you will be assessed on your understanding of AI concepts, technical implementation, and ability to create an interactive application.

**Problem Statement:**

**Our institution currently faces challenges in providing timely, accurate, and efficient support to students, faculty, and staff. The existing support systems, such as help desks and FAQs, are often overwhelmed, leading to long wait times, information overload, and decreased user satisfaction.**

**Modules:**

1. Prepare Data
2. Process the Data using NLTK for the model training
3. Define the model structure (Feed Forwarding Neural Network) of deep learning
4. Prepare the Dataset based on processed data
5. Train the model & Save the Trained model
6. Create a Chat-Bot based on a trained model

**About Data**:

Data format: JSON

Data contains: University Data of MKCE, Karur.

**Project Approach :**

**Model Training Approach**

**1. Text Preprocessing**

**nltk\_process Class:** This class provides functions for text preprocessing:

* + tokenize(sentence): Splits a sentence into individual words (tokens).
  + stem(word): Reduces a word to its root form using Porter stemming (improves word similarity).
  + bag\_of\_words(tokenized\_sentence, all\_words): Converts a tokenized sentence into a numerical representation (bag-of-words) based on a vocabulary of all words.

**2**. **Neural Network Model**

**NeuralNet Class:** This class defines a feed-forward neural network architecture with:

* + Input layer size: Determined by the length of the bag-of-words features.
  + Hidden layer size: Set as a hyper- parameter (currently hidden\_size = 7).
  + Output layer size: Determined by the number of unique intent labels.
  + Activation function: ReLU (Rectified Linear Unit) is used for hidden layers.

**3**. **Dataset Preparation**

**ChatDataset Class:** This class creates a PyTorch Dataset object for training the model.

It holds the training data (x\_train - features, y\_train - labels) in memory.

**4**. **Data Processing for Training**

The code snippet demonstrates how the training data is prepared:

**Vocabulary Creation:**

* + Extracts intents and patterns from the intents dictionary.
  + Creates a list of all unique words (all\_words) by tokenizing patterns and filtering out stop words.

**Feature and Label Creation:**

* + Iterates through each pattern-intent pair:

Converts the pattern sentence into a bag-of-words representation using the nltk\_process.bag\_of\_words function.

Look up the corresponding intent label index from the unique intent list (tags).

**Data Conversion:**

* + Converts both features (x\_train) and labels (y\_train) into NumPy arrays for efficient model training.

**5. Hyperparameters**

num\_epochs: Number of training iterations (currently set to 1000).

batch\_size: Number of training examples used per model update (currently set to 7).

learning\_rate: Controls the step size during model optimization (currently set to 0.001).

input\_size: Determined dynamically based on the bag-of-words feature size.

hidden\_size Hyperparameter defining the hidden layer's size (currently set to 7).

output\_size: Determined dynamically based on the number of unique intent labels.

**6. Training Process**

The code defines a training loop that iterates for num\_epochs.

Within each epoch, it iterates through batches of data from the train\_loader.

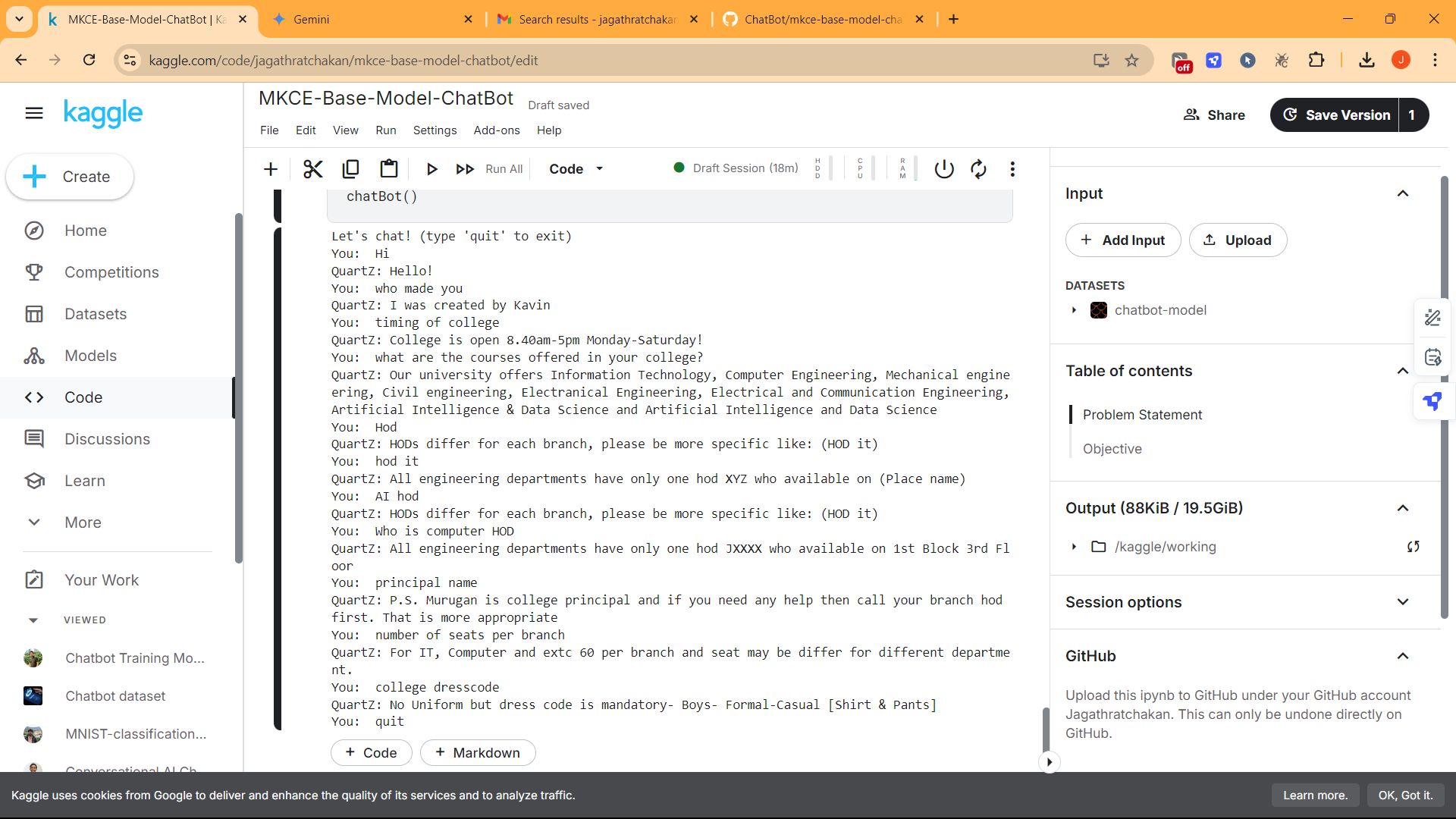
For each batch:

* + Moves data (words and labels) to the appropriate device.
  + Performs a forward pass through the neural network to obtain model predictions.
  + The cross-entropy loss function calculates the loss (error) between predictions and actual labels.
  + Performs back-propagation to update the model weights based on the calculated loss.

Challenges Faced and How I Overcome:

Initially struggled to collect the data for this project but later I got some relevant data from online and helped to understand how that data works in the chatbot in a certain format.

Also, I faced challenges while training the model on the laptop that time I spent more time learning about the GPU version of the laptop and how to update it to the latest version which showed an error while creating a model. But I failed to update the version at that time I got an idea to train the model with a different environment. Platforms tested for the model training in my laptop were Jupyter Notebook, collab, and Kaggle. I got a successful outcome in the Kaggle so only I push a Kaggle source code in the GitHub.

**Outcome**