CSCI 6517 Recommender Systems
Lab and Assignment 4: Retrieval Augmented Generation

July 20, 2025

This lab and assignment involves creating a recommender engine using Retrieval Augmented Generation, to improve performance using large language models.

- Programming language: Python (Jupyter IPython Environment)
- Due Date: Posted in Syllabus (Aug 1st, 2025)

Marking scheme and requirements: Full marks will be given for (1) working, readable, reasonably efficient, documented code that achieves the assignment goals and (2) for providing appropriate answers to the questions in a Jupyter notebook (named assignment-rag.ipynb).

Please adhere to the collaboration policy on the course website – people you discussed the assignment solution with, or websites with source code you used should be listed in the submitted Jupyter notebook.

What/how to submit your work:

- All your plot, result tables, and answers to the questions should be included in a notebook named assignment-rag.ipynb.
- Submit your ipython notebook (including the PDF version of it) to BrightSpace. You must include the outputs of each cell in the submitted version. Your last submission before the assignment deadline will be considered to be your submission.

1 Before You Begin

In this assignment you'll not be given any code base to complete. You have to use the existing tutorial codes and internet as your resource for completing the notebook.

The recommendation dataset we will be using is from Flipkart. This dataset contains several products that can be used with their title, description, price, ratings etc. This anonymized dataset contains a sample of 20k product data from a bigger dataset of 5.8M.

You will also need to download the Flipkart data file from kaggle website (https://www.kaggle.com/datasets/PromptCloudHQ/flipkart-products) and upload it into the same folder where you implement your codebase.

The weight of each question is provided on the right of each question. Total weight is- **45.**

You are free to create new cells as much as needed following the convention in the provided notebook as template. Note: please maintain the formatting of the notebook for ease of marking.

2 Main Assignment

Please answer the questions below and provide IPython implementations.

Information: You are going to use a **LLAMA-3** model for completing your assignment. You are free to use any library and vector index for your assignment. However, *your notebook should be self-sufficient and should run sequentially* when marker marks your assignment without any change.

Q1. Implement RAG based Product Recommendation System

For the evaluation of your assignment, your notebook is divided into several parts. Your implementation should be placed in the appropriate sections. For each part of your implementation, *provide a description explaining the rationale and application of each code section of your code for evaluation purposes.* Lack of description might indicate a blind copy of code without understanding, which could result in a loss of points.

(a) Import Necessary Libraries

(b) Data Pre-processing and Preparation: Load, Prepare and Pre-process data. Decide what information you will need for your recommendations. (10)

(c) Implementing Retrieval Engine: Index the documents and retrieval engine. Show intermediate outputs of these portions to check the modules are working. (8)

(d) Implementing Generation: Implement generation using a LLAMA3 model. (8)

(e) Output and Demonstration: Design prompt(s) and query. Show at least 3 output demonstrations and top-3 relevant products for each demonstration. (5)

Q2. Answer the following questions

(a) In Data processing, which information did you retain and which ones you dropped? Why? (2)

(b) In the retrieval, did you use any textual retrieval algorithm? Can you explain why you should and/or should not? (2)

(c) Why do we need a model to embed? Should you use the same model for embedding documents and generation? Why or why not? (2)

(d) Did you face any problem using the LLAMA model? If yes, what measure did you take to use the model? If not, why do you think you did not? (2)

(e) Language models have limitation of context size. What is the context size of the model you use? Could you meet the requirement? (2)

(f) Can you suggest how we can personalize the recommendation for each user using RAG? (2)

(g) In this assignment, no performance evaluation was not asked. However, to evaluate the performance, how would you approach? (2)