Technical Report Advanced AI

Introduction

When people only have a few ingredients left at home, they often struggle to come up with a recipe and don’t want to go to the store or order takeout. This project aims to solve that problem by providing an AI-powered recipe generator that takes a list of available ingredients and produces complete, creative recipes.

As someone who lacks creativity in the kitchen, this idea was personally appealing. I often find myself unsure of what to cook, even when I have several items in my fridge. This project offers a solution that could help both me and others in similar situations.

The chosen approach was to build a Streamlit web app powered by Ollama, a local AI model that generates recipes without needing an internet connection. The app allows users to input ingredients and then displays three unique recipes generated in a moments notice.

GitHub Repository:  
<https://github.com/ED-ISA-243/AdvancedAI_IndividualProject>

# Data

# This project does not use a traditional dataset. Instead, it relies on user-provided input ( a list of ingredients entered at runtime). The application uses these ingredients to create prompts that are passed to the AI model. Because of this setup, no preprocessing steps like data cleaning or transformation were necessary. The “data” is generated dynamically based on user input, making each prompt and response unique. The main challenge was not data processing but ensuring valid and relevant AI output.

# Model & Methods

Initially, I explored various model options:

* Self-made models: I first tried a XGBoost model, of which I soon realized classification is not good for this project, since I needed the model to write me recipes instead of just showing which ingredients classified the best to any recipe. After that I decided to go for already existing models.
* OpenAI models: These had drawbacks like API limitations and token cutoffs, which often caused the model to stop mid-recipe and was in general annoying to work with. However, the parts that did get generated were rather accurate and usable.
* Locally downloaded models: These required a lot of memory (over 30GB RAM) and were slow or inaccurate. I first tried GPT-J-6B by EleutherAI, which has 6B parameters and took very long to generate inaccurate recipes. Then I switched to GPT-Neo-1.3B, which is also made by EleutherAI. This model was more efficient since it was way lighter than the previous model, but still not good enough to rely on since the recipes often did not make any sense.

Eventually, I discovered Ollama, a platform that also lets users run AI models locally with optimized performance. I used llama3.2:latest for this project.  
It provided:

* Fast generation times
* Offline access with 0 limitations (no API keys or rate limits)
* Reliable, complete, and especially coherent recipe outputs

To maintain quality, I implemented:

* A custom prompt instructing the AI to generate recipes using only the listed ingredients
* A validation filter (is\_valid\_recipe) to screen out blog-like or promotional text, since the model initially did do that
* A retry mechanism that makes up to three attempts if the output is invalid

# **Results & Evaluation**

The app generates three full recipes based on at least three user-entered ingredients. Each recipe includes a relevant title, a list of ingredients with generated quantities (if you don’t give quantities yourself) and a minimum of five clear steps how to make the recipe.

The results are very interesting. The first models I tried took several minutes to generate poor recipes, while this one generates it in around 10 seconds on my laptop with a **creative**, **realistic** and **coherent** outcome. These 3 were the main criteria I wanted to succeed in.

# Contributions

I have worked on this project alone, with using GenAI as my copilot. Inspiration, model suggestions, method of approach, correction of code were all done and aided by AI. I have tried to look up models that would fit my project online or tried looking up methods of approach on how to start with this project, but the results were poor. AI helped where normal google searches could not.  
I have also used AI to generate a cheat sheet on all commands/implementations for Streamlit, a package that requires specific python commands I had 0 knowledge about.

# Challenges & Future work

I found it very challenging to decide which model to use. I first tried to make a model myself, but that ended up to be a dead end since the model I generated was very flawed and not goal fulfilling.

I could improve by implementing a filter that filters for example vegetarian options or gluten-free recipes etc. The application itself could also be cleaner, better and in general more appealing. I just made a quick Streamlit application that does the task perfectly, but it could look better.