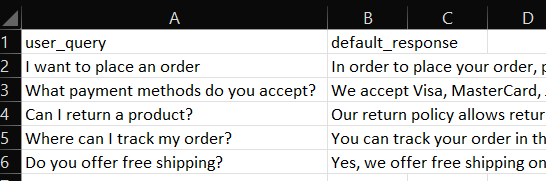
**ChatBot - resolving general queries (Warda)**

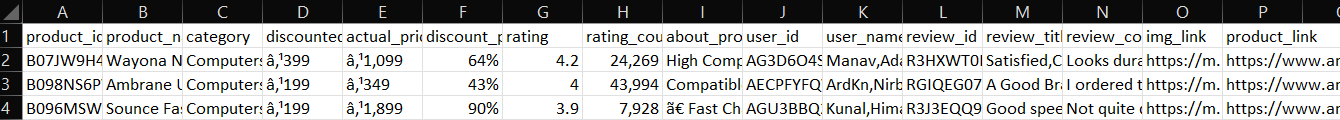
I am working on a chatbot that handles basic queries about the products. It can handle predefined user queries as well as displaying default answers that might be included in the company’s business rule requirements.

There are two CSV files that the agent is reading data from using **Panda’s data frames**.

1. **default\_response.csv** file, that the agent checks for predefined user queries.



1. **amazon.csv** file, which contains all the details about the products and user reviews.



The chatbot is set up using **GPT-4 model**, which interacts with an agent that has read the provided data frames.

Input from user is being taken continuously until the users quits. During this loop the query is being processed, where the query is first compared with the pre-defined users query in default\_response.csv file. The query is being compared using the **SpaCy similarity function** which computes the similarity score, if the score is greater than 0.5 default response is displayed by the bot. The range of 0.5 can be changed as per our need.

If the queries match, default response is displayed else it then searches through the amazon.csv file to see if the user query can be solved through the data provided about the products in the excel sheet.

**LangChain framework** here helps to instantiate the chatbot and to create an agent that utilizes Pandas Data Frames for interaction with the chatbot. It helps simplify the setup and integration of OpenAI GPT-4 model and Pandas Data Frame agent.

I have also introduced **caching concept**. Generated responses from the bot will be stored in the cache to avoid any redundant computations for repeated user queries. If the user asks the same query, bot response will be faster.