# LLM\_pipeline function:

The selected code defines a function llm\_pipeline that appears to be part of a larger process for handling and processing documents using a language model. Here's a detailed explanation of the selected code:

**Function Definition**

def llm\_pipeline(file\_paht):

This line defines a function named llm\_pipeline that takes a single parameter file\_paht. Note that there is a typo in the parameter name; it should likely be file\_path.

**Calling `file\_processsing`**

document\_ques\_gen, document\_answer\_gen = file\_processsing(file\_paht)

This line calls the file\_processsing function (which also has a typo and should be file\_processing) with the provided file\_paht argument.

The function file\_processsing is expected to return two values: document\_ques\_gen and document\_answer\_gen. However, the file\_processsing function as defined earlier does not return any values, which will cause an error.

**Initializing ChatOpenAI**

llm\_ques\_gen\_pipeline = ChatOpenAI(

temperature=0.3,

model='gpt-3.5-turbo'

)

This initializes an instance of ChatOpenAI with a specified temperature of 0.3 and the model gpt-3.5-turbo.

ChatOpenAI is likely a class from the langchain library used to interact with the OpenAI language model.

**Creating Prompt Templates**

PROMPT\_QUESSIONS = PromptTemplate(template = prompt\_template, input\_variables=["text"])

This line creates a PromptTemplate instance named PROMPT\_QUESSIONS using a template prompt\_template and specifies that the input variable for this template is text.

prompt\_template should be defined elsewhere in the code.

refined\_prompt\_questions = PromptTemplate(template=refine\_template, input\_variables=["existing\_answers", "text"])