Terminal 1   ✕   orchestration_llms_with_lang ●   +

```python
## Task 2. Install packages and import libraries
```

```python
# Install Vertex AI SDK & Other dependencies

!sudo apt -y -qq install tesseract-ocr
!sudo apt -y -qq install libtesseract-dev
!sudo apt-get -y -qq install poppler-utils #required by PyPDF2 for page count and other pdf utilities
!sudo apt-get -y -qq install python-dev libxml2-dev libxslt1-dev antiword unrtf poppler-utils tesseract-ocr flac ffmpeg lame libmad0 libsox-fmt-mp3 sox lib
!pip install langchain langchain-community langchain-core langchain-google-vertexai pypdf
```

```python
# Automatically restart kernel after installs so that your environment can access the new packages
import IPython

app = IPython.Application.instance()

app.kernel.do_shutdown(True)
```

```python
# Import libraries

import urllib
import warnings
from pathlib import Path as p

import pandas as pd
from langchain import PromptTemplate
from langchain.chains.summarize import load_summarize_chain
from langchain.document_loaders import PyPDFLoader
from langchain_google_vertexai import ChatVertexAI

warnings.filterwarnings("ignore")
```

Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy   Yes   No

```python
## Task 3. Load the model and prepare data
```

```python
#TODO: Load the pre-trained text generation model named "gemini-1.5-pro" using "ChatVertexAI" class.
```

```python
#TODO: Download a PDF file from specified URL and save it in "data" directory.
```

```python
#TODO: Load the PDF file and split it into individual pages.
```

```python
## Task 4. Generate summaries
```

```python
#Prompt design with Stuffing Chain

prompt_template = """Write a concise summary of the following text delimited by triple backquotes.
              Return your response in bullet points which covers the key points of the text.
              ```{text}```
              BULLET POINT SUMMARY:
  """

prompt = PromptTemplate(template=prompt_template, input_variables=["text"])
```

```python
#TODO: Set up a summarization chain using the stuff method. Incorporate the model loaded earlier into the summarization c
```

Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy   Yes

```python
## Task 5. Add tools to the LLM for multiplication
```

```python
#TODO: Create a tool called 'multiply' that takes two integers, 'a' and 'b', as inputs, performs the multiplication (a * b), and returns the result.
from langchain_core.tools import tool
```

```python
#TODO: Add the tools to the LLM created earlier and invoke it with the following query. Print the result in the console.
from langchain_core.messages import HumanMessage, ToolMessage, AIMessage
query = "What is 3 * 12?"
messages = [HumanMessage(query)]
```

```python
#TODO: Iterate through the tools in the response, invoke the tools and append the response to the messages object.
```

```python
#TODO: Invoke the LLM with the solution of the tool and the original message and print the final user response.
```

Simple   1   1   Python 3 (ipykernel) (Local) | Idle   Mode: Command   Ln 1, Col 1   orchestration_llms_with_langchain_challenge_notebook-v1.0.0.ipynb   2