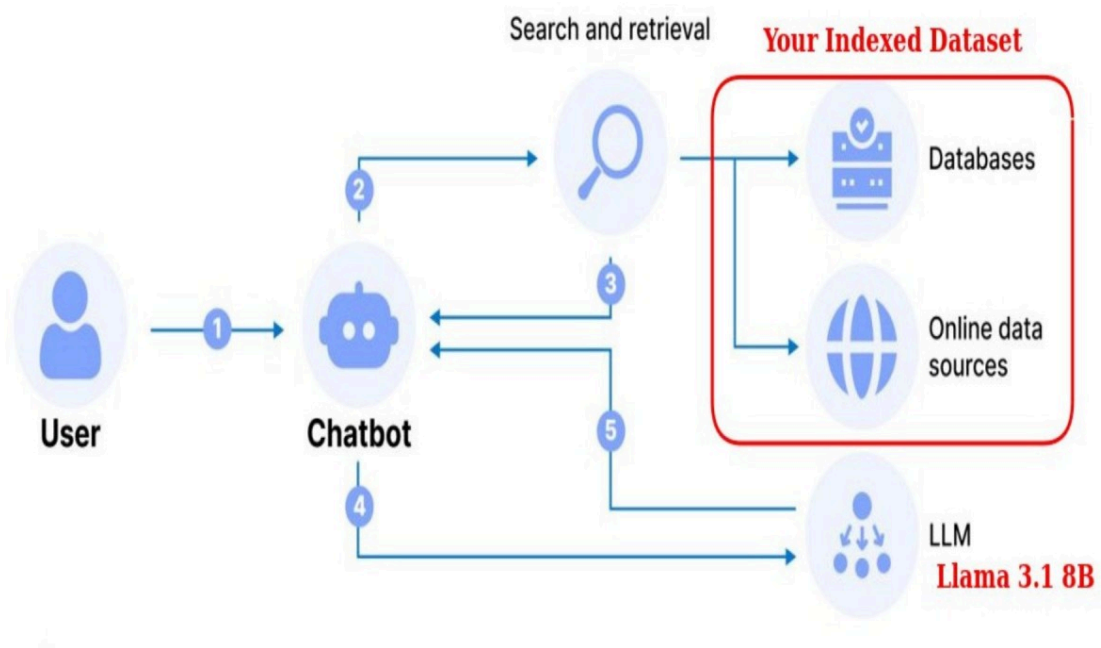


Build AI RAG Chatbot with Ollama and LangChain

Mohammad Oghli



Mohammad Oghli

Software Lead @ Archireef | Data Solutions | MLOps | Tech Author

Published Aug 29, 2024

[+ Follow](#)

Today I will demonstrate in this article how to build your own AI chatbot on your customized dataset using Retrieval-Augmented Generation (RAG).

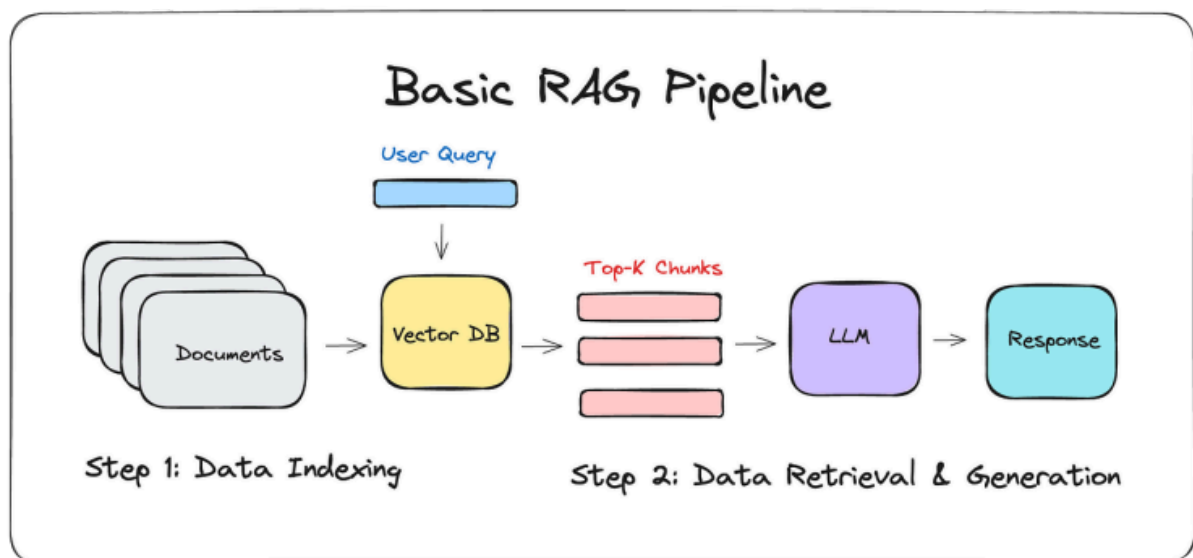
We will use to develop the RAG chatbot: **Ollama** to run the Llama 3.1 LLM locally on your device and **LangChain** framework to build chatbot application.

First of all what is Retrieval-Augmented Generation (RAG) ?

RAG is AI development technique where a large language model (LLM) is connected to an external knowledge base to improve the accuracy and quality of its responses.

In brief RAG technique allow us to feed specific dataset to the LLM and ask questions about it.

Basic RAG Chatbot Pipeline



RAG Pipeline

There are 2 basic steps we need to follow in order to create RAG chatbot:

- First we should index our dataset into Vector store: here we feed specific data documents after applying some operations on it to embed it and store it in vector store.
- After that we can query the indexed dataset in vector store and retrieve relevant information about it and then LLM generates response based on user prompt and the retrieval information of indexed dataset.

Build RAG Chatbot Workflow

We can divide our work into these sections:

- Install Ollama and download Llama 3.1 8B on your device to run it locally
- Prepare the customized dataset that will be feed to LLM (Llama 3.1)
- Index the dataset into Vector store using [LangChain](#)
- Setup Llama 3.1 LLM with LangChain and integrate our indexed dataset query to the LLM prompt

Install Ollama and run Llama 3.1 locally

we will use [Ollama](#) platform to download and run open source LLMs locally on our device. In our project we will use **Llama 3.1 8B** by Meta.

- First we should install Ollama on our device

To install it on **Linux** OS Run this command:

```
curl -fsSL https://ollama.com/install.sh | sh
```

if you are using Windows or macOS you can check [installation section here](#).

- After installing Ollama we can now download and run any of the following open source models locally on our device

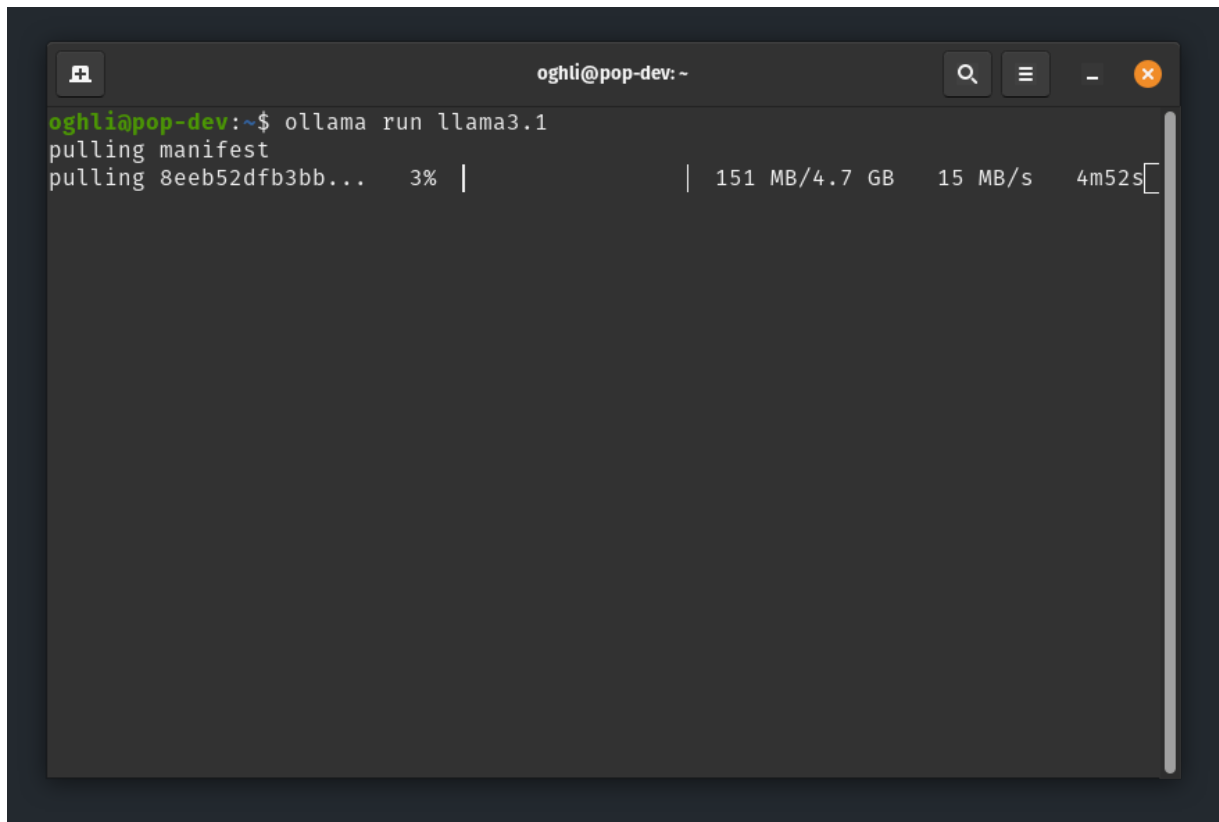
Model	Parameters	Size	Download
Llama 3.1	8B	4.7GB	<code>ollama run llama3.1</code>
Llama 3.1	70B	40GB	<code>ollama run llama3.1:70b</code>
Llama 3.1	405B	231GB	<code>ollama run llama3.1:405b</code>
Phi 3 Mini	3.8B	2.3GB	<code>ollama run phi3</code>
Phi 3 Medium	14B	7.9GB	<code>ollama run phi3:medium</code>
Gemma 2	2B	1.6GB	<code>ollama run gemma2:2b</code>
Gemma 2	9B	5.5GB	<code>ollama run gemma2</code>
Gemma 2	27B	16GB	<code>ollama run gemma2:27b</code>
Mistral	7B	4.1GB	<code>ollama run mistral</code>
Moondream 2	1.4B	829MB	<code>ollama run moondream</code>
Neural Chat	7B	4.1GB	<code>ollama run neural-chat</code>
Starling	7B	4.1GB	<code>ollama run starling-lm</code>
Code Llama	7B	3.8GB	<code>ollama run codellama</code>
Llama 2 Uncensored	7B	3.8GB	<code>ollama run llama2-uncensored</code>
LLaVA	7B	4.5GB	<code>ollama run llava</code>
Solar	10.7B	6.1GB	<code>ollama run solar</code>

Ollama Models

we will download and install **Llama 3.1 8B** parameters which require at least **4.7 GB** hard disk space and **8 GB of RAM** available to run the model.

Run this command to download Llama 3.1 model on your device

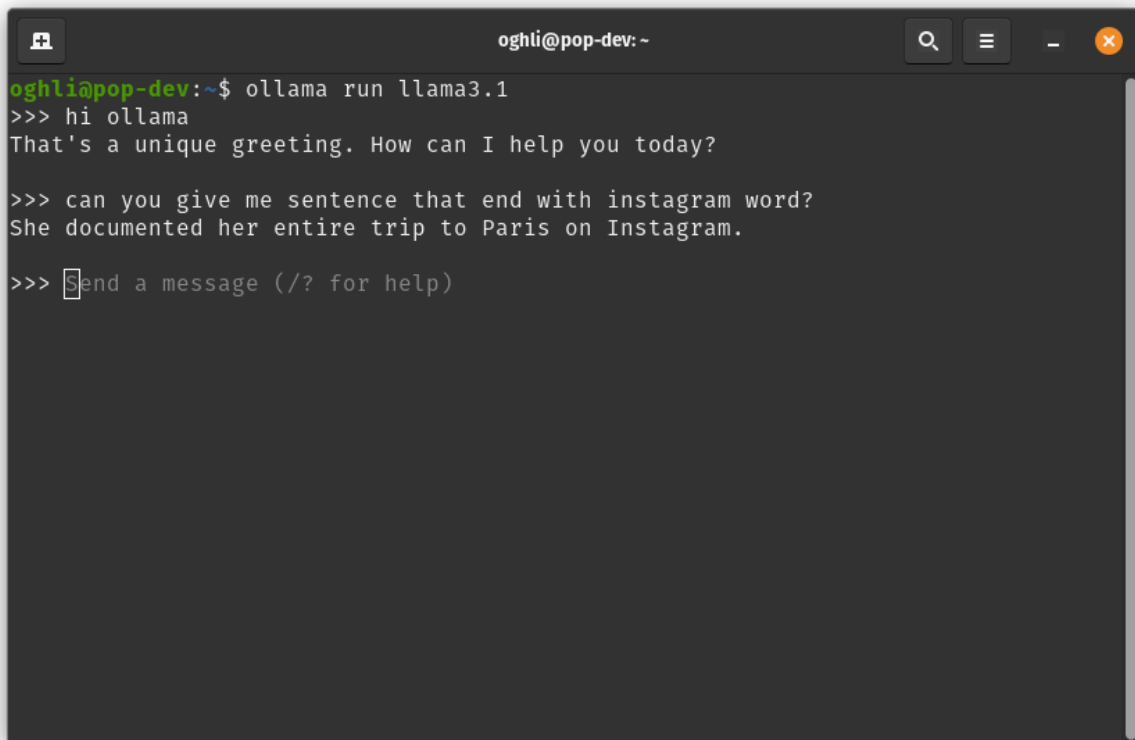
```
ollama run llama3.1
```



```
oghli@pop-dev: ~  
oghli@pop-dev:~$ ollama run llama3.1  
pulling manifest  
pulling 8eeb52dfb3bb... 3% | | 151 MB/4.7 GB 15 MB/s 4m52s
```

Ollama Llama 3.1

Then after the model download completed it will start automatically in your terminal and you can chat with it, to get out from chat press **CTRL+D**

A terminal window titled 'oghli@pop-dev: ~' with search, menu, and window control icons in the title bar. The terminal shows the command 'ollama run llama3.1' being executed. The user then interacts with the model by sending two prompts: 'hi ollama' and 'can you give me sentence that end with instagram word?'. The model responds with 'That's a unique greeting. How can I help you today?' and 'She documented her entire trip to Paris on Instagram.' respectively. The terminal ends with a prompt 'Send a message (/? for help)'.

```
oghli@pop-dev:~$ ollama run llama3.1
>>> hi ollama
That's a unique greeting. How can I help you today?

>>> can you give me sentence that end with instagram word?
She documented her entire trip to Paris on Instagram.

>>> Send a message (/? for help)
```

Llama 3.1

you can check its meta information by running

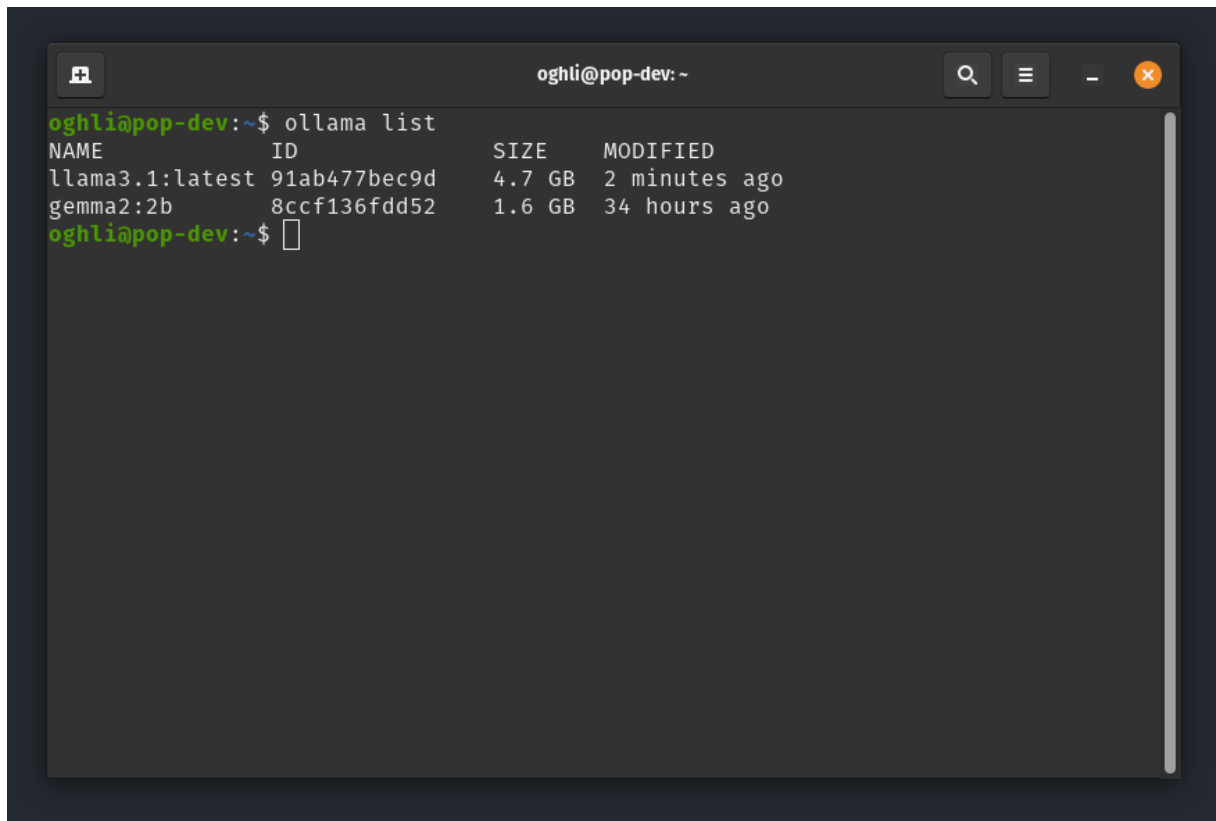
```
ollama show llama3.1
```

```
oghli@pop-dev: ~  
oghli@pop-dev:~$ ollama show llama3.1  
Model  
  arch          llama  
  parameters    8.0B  
  quantization  Q4_0  
  context length 131072  
  embedding length 4096  
  
Parameters  
  stop "<|start_header_id|>"  
  stop "<|end_header_id|>"  
  stop "<|eot_id|>"  
  
License  
  LLAMA 3.1 COMMUNITY LICENSE AGREEMENT  
  Llama 3.1 Version Release Date: July 23, 2024  
  
oghli@pop-dev:~$
```

Llama 3.1 Info

Also you can show list of downloaded models on your device

```
ollama list
```

A terminal window titled 'oghli@pop-dev: ~' with search, menu, and window control icons in the title bar. The terminal shows the command 'ollama list' and its output, which is a table with four columns: NAME, ID, SIZE, and MODIFIED. The output lists two models: 'llama3.1:latest' and 'gemma2:2b'.

```
oghli@pop-dev:~$ ollama list
NAME                ID                SIZE    MODIFIED
llama3.1:latest     91ab477bec9d     4.7 GB  2 minutes ago
gemma2:2b           8ccf136fdd52     1.6 GB  34 hours ago
oghli@pop-dev:~$
```

Ollama List

Prepare the customized dataset for RAG on Llama 3.1

At this step we will use this public dataset [AI-Powered Job Market Insights](#) on Kaggle platform.

The dataset contains the following columns attributes:

- Job_Title
- Industry
- Company_Size
- Location
- AI_Adoption_Level
- Automation_Risk
- Required_Skills
- Salary_USD
- Remote_Friendly
- Job_Growth_Projection

About this file Add Suggestion									
This file does not have a description yet.									
A Job_Title		A Industry		A Company_Size		A Location		A AI_Adoption_Level	
categorical		categorical		categorical		string		categorical	
Data Scientist	12%	Manufacturing	12%	Small	34%	San Francisco	12%	Medium	36%
HR Manager	11%	Education	11%	Large	33%	Singapore	11%	Low	35%
Other (381)	76%	Other (385)	77%	Other (163)	33%	Other (384)	77%	Other (147)	29%
Cybersecurity Analyst		Entertainment		Small		Dubai		Medium	
Marketing Specialist		Technology		Large		Singapore		Medium	
AI Researcher		Technology		Large		Singapore		Medium	
Sales Manager		Retail		Small		Berlin		Low	
Cybersecurity Analyst		Entertainment		Small		Tokyo		Low	
UX Designer		Education		Large		San Francisco		Medium	
HR Manager		Finance		Medium		Singapore		Low	
Cybersecurity Analyst		Technology		Small		Dubai		Medium	
AI Researcher		Retail		Large		London		High	
Sales Manager		Entertainment		Medium		Singapore		High	
Marketing Specialist		Finance		Small		London		High	
AI Researcher		Entertainment		Medium		Singapore		Medium	

Kaggle Dataset

We will take only the **first 25 records** of this dataset for the project demo to minimize the processing time of embedding and indexing the dataset in Vector store locally.

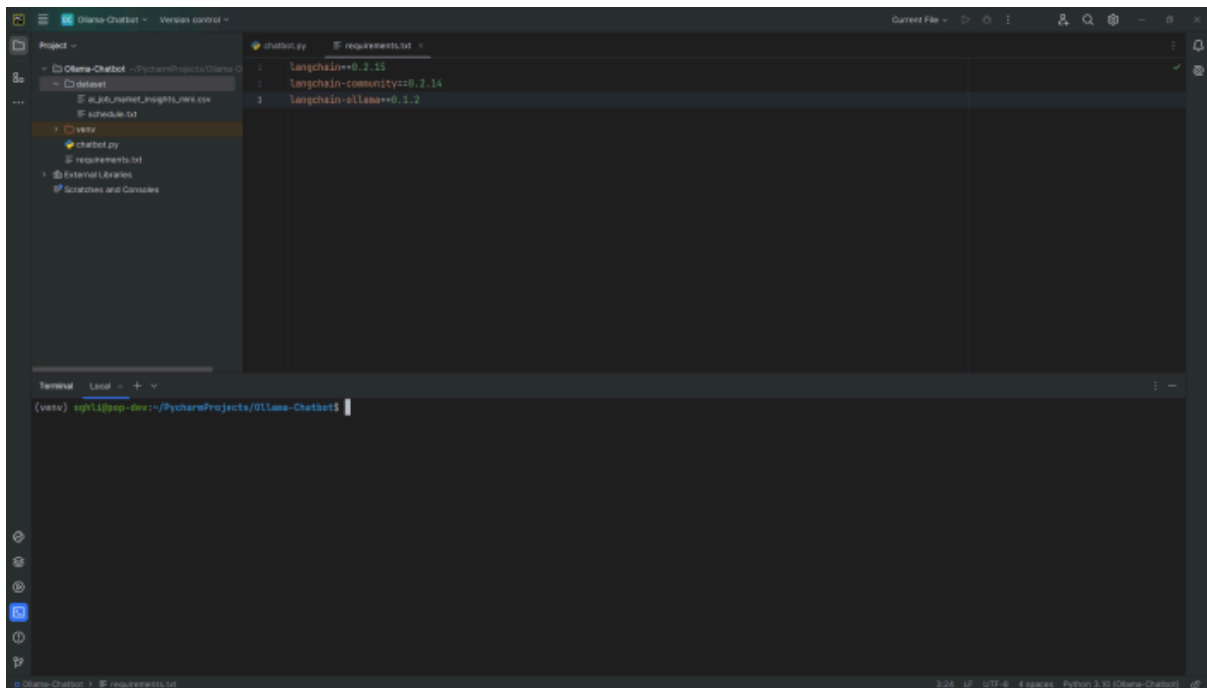
You can download the dataset from [Kaggle](#) and save only the first 25 records into CSV file.

Job Title	Industry	Company Size	Location	AI Adoption Level	Automation Risk	Required Skills	Salary USD	Remote Friendly	Job Growth Projection
Cybersecurity Analyst	Entertainment	Small	Dubai	Medium	High	UX/UI Design	111382.180345018	Yes	Growth
Marketing Specialist	Technology	Large	Singapore	Medium	High	Marketing	83792.5624061001	No	Decline
AI Researcher	Technology	Large	Singapore	Medium	High	UX/UI Design	107170.263606805	Yes	Growth
Sales Manager	Retail	Small	Berlin	Low	High	Project Management	86027.8607073842	No	Growth
Cybersecurity Analyst	Entertainment	Small	Tokyo	Low	Low	JavaScript	87792.8023709024	Yes	Decline
UX Designer	Education	Large	San Francisco	Medium	Medium	Cybersecurity	302029.902809508	No	Growth
HR Manager	Finance	Medium	Singapore	Low	High	Sales	202005.7226734740	Yes	Growth
Cybersecurity Analyst	Technology	Small	Dubai	Medium	Low	Machine Learning	86607.3178180969	Yes	Decline
AI Researcher	Retail	Large	London	High	Low	JavaScript	76055.8808437189	No	Stable
Sales Manager	Entertainment	Medium	Singapore	High	Low	Cybersecurity	86034.8702045007	Yes	Decline
Marketing Specialist	Finance	Small	London	High	High	Python	91586.9703702510	Yes	Growth
AI Researcher	Entertainment	Medium	Singapore	Medium	High	UX/UI Design	76902.5802443849	No	Growth
AI Researcher	Transportation	Large	San Francisco	High	Medium	Python	720551.8803480541	Yes	Growth
HR Manager	Retail	Small	Paris	Low	High	Cybersecurity	86036.5168758891	Yes	Decline
Product Manager	Finance	Medium	Singapore	High	High	JavaScript	303305.901142954	Yes	Growth
Sales Manager	Telecommunications	Small	Dubai	High	Low	UX/UI Design	82799.3577086020	No	Growth
Software Engineer	Manufacturing	Medium	Singapore	High	Medium	Machine Learning	79401.2500017415	No	Decline
AI Researcher	Manufacturing	Large	Tokyo	Low	High	Project Management	73284.8962181562	No	Stable
Product Manager	Finance	Large	San Francisco	Medium	High	UX/UI Design	97946.0807020894	Yes	Stable
Software Engineer	Entertainment	Small	Dubai	Low	Low	UX/UI Design	73820.4505271853	Yes	Growth
Sales Manager	Retail	Medium	Sydney	Medium	Low	JavaScript	124422.892380415	Yes	Decline
Data Scientist	Manufacturing	Small	Dubai	Medium	High	JavaScript	81791.9713695001	Yes	Growth
UX Designer	Retail	Small	Tokyo	High	Medium	Cybersecurity	61898.7144425480	No	Decline
Product Manager	Manufacturing	Medium	Paris	Medium	Medium	Data Analysis	80396.5880074173	Yes	Decline
Operations Manager	Transportation	Medium	San Francisco	Low	Low	UX/UI Design	88530.3304000239	Yes	Stable
Data Scientist	Entertainment	Medium	Sydney	Low	Low	Sales	112234.37330484	Yes	Stable
UX Designer	Transportation	Large	Tokyo	Medium	High	UX/UI Design	85762.5247345541	No	Decline
HR Manager	Manufacturing	Small	Tokyo	High	Low	Cybersecurity	73823.296311204	Yes	Growth
Cybersecurity Analyst	Telecommunications	Small	New York	High	High	Marketing	84272.1823000645	Yes	Stable
Marketing Specialist	Telecommunications	Large	San Francisco	Low	Medium	Python	86548.3954330186	Yes	Growth
Operations Manager	Transportation	Medium	Dubai	High	Medium	Project Management	40705.7149617616	Yes	Stable
HR Manager	Telecommunications	Large	London	Medium	High	Data Analysis	116772.438687609	Yes	Stable
Sales Manager	Manufacturing	Large	Paris	Medium	Low	Machine Learning	73376.602038087	No	Decline
HR Manager	Finance	Large	London	Low	Medium	Machine Learning	110340.878430902	Yes	Stable
Cybersecurity Analyst	Transportation	Small	Dubai	Medium	Low	Machine Learning	87946.3628253116	Yes	Stable
Sales Manager	Healthcare	Large	Dubai	Low	Medium	JavaScript	102581.68272788	Yes	Growth
Operations Manager	Entertainment	Small	San Francisco	Medium	High	Machine Learning	317868.884750025	Yes	Stable
Cybersecurity Analyst	Telecommunications	Small	Dubai	Low	High	Project Management	101482.64193389	Yes	Stable
Software Engineer	Telecommunications	Small	New York	High	High	Python	303457.724320349	Yes	Stable
Marketing Specialist	Technology	Medium	Sydney	Medium	Medium	Machine Learning	86711.671267724	Yes	Growth
Operations Manager	Education	Small	Dubai	Medium	Low	UX/UI Design	87643.3758264619	Yes	Decline
Software Engineer	Transportation	Small	London	Low	Low	UX/UI Design	76790.043008052	Yes	Decline
UX Designer	Energy	Medium	Dubai	Low	Medium	JavaScript	80331.770003407	Yes	Decline
Operations Manager	Telecommunications	Small	New York	Medium	Medium	Project Management	128203.900993176	No	Stable
UX Designer	Entertainment	Large	Singapore	Low	Medium	Python	73636.1262499616	No	Decline
Sales Manager	Healthcare	Large	Singapore	Medium	Medium	Data Analysis	64886.1634327256	No	Growth
Software Engineer	Telecommunications	Small	Tokyo	High	Medium	Cybersecurity	82389.3807343099	Yes	Growth
Marketing Specialist	Healthcare	Medium	Paris	Low	Medium	Python	83794.910544304	Yes	Stable

Customized Dataset

Index the dataset into vector store using LangChain

First we should setup our project and install python packages dependencies in our virtual environment



Project Structure

You can view the project structure here:

- **dataset/** directory for the data we need to index
- **chatbot.py** python script for the RAG chatbot application
- **requirements.txt** for project python packages

The requirements file contains the following packages

```
langchain==0.2.15
```

```
langchain-community==0.2.14
```

```
langchain-ollama==0.1.2
```

To install these python packages in your virtual environment run

```
pip install -r requirements.txt
```

Now we can load our dataset and index it into a vector store

We can load the dataset using **TextLoader** method which allow us to load any document file for different data formats such as text (**.txt**), spread sheets (**.csv**) ...etc

```
from langchain_community.document_loaders import  
TextLoader
```

```
# Create a TextLoader object
```

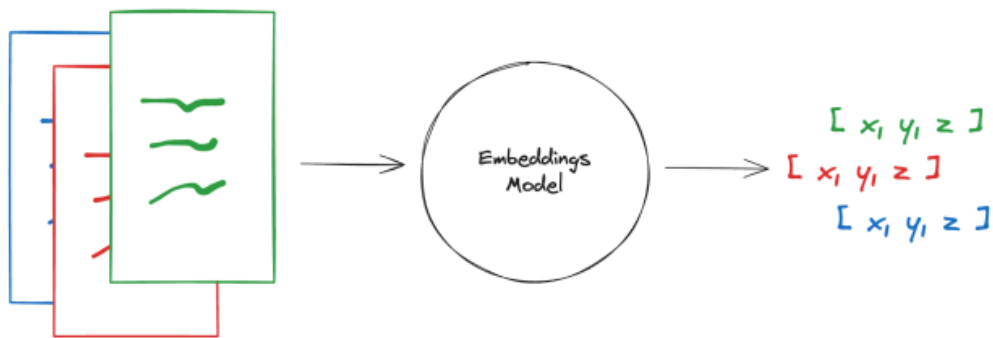
```
loader =  
TextLoader("dataset/ai_job_market_insights_mini.csv")
```

Here we load our **ai_job_market_insights_mini.csv** data sheet

Then we need to create embedding object that will be used to index our dataset into vector store. In order to create the object we need to use specific embedding model.

What is Embedding models?

Embedding models create a vector representation of a piece of text. You can think of a vector as an array of numbers that captures the semantic meaning of the text.



Embedding Model

In our project we will use **OllamaEmbeddings** model

```
from langchain_community.embeddings import  
OllamaEmbeddings
```

```
# Create an OllamaEmbeddings object
```

```
embeddings = OllamaEmbeddings(model="llama3.1")
```

Then we should use **VectorstoreIndexCreator** wrapper method to start indexing our dataset into vector store. this method by default will use **InMemoryVectorStore** which will store the dataset index in your device memory during running RAG chatbot and doesn't need setting up. you can also use other vector databases such as **Chroma** but you have to setup the vector database before running the application.

In this step we will create **VectorstoreIndexCreator** object and then index our loaded dataset into the vector store

```
from langchain.indexes import VectorstoreIndexCreator
```

```
# Create a VectorstoreIndexCreator object
```

```
index_creator =
```

```
VectorstoreIndexCreator(embedding=embeddings)
```

```
# Call from_loaders method
```

```
index = index_creator.from_loaders([loader])
```

```
print("indexing document in vector store completed!")
```

The indexing operation will take some time depending on the size of your dataset and the computing resources available in your device.

Setup Llama 3.1 LLM and integrate our indexed dataset with it

Finally, after we finished indexing our dataset into vector store we should setup the LLM that we will use it in our chatbot application.

We will create **ChatOllama** object and set LLM to **llama3.1** which we already installed locally on our device using **Ollama**

```
from langchain_ollama import ChatOllama
```

```
# Create a ChatOllama object
```

```
chat_llama3 = ChatOllama(model="llama3.1",  
temperature=0.7)
```

Then we only need to send prompt query to it using the vector store index object that we created for our dataset

```
answer = index.query(prompt, llm=chat_llama3)
```

Now we can start testing our RAG chatbot application and ask it questions related to our dataset content

```
prompt = "Can you list job title for only technology  
industry?"
```

```
answer = index.query(prompt, llm=chat_llama3)
```

The RAG chatbot will generate the following response based on the dataset information provided to it

Here are the job titles from the provided context that belong to the Technology industry:

1. Marketing Specialist

2. AI Researcher (x3)

To streamline the process of prompting the RAG chatbot we can run the following loop

```
prompt = ""
while prompt.lower() != "exit":
    # Use ChatOllama object to answer questions
    prompt = input("Enter your query: ")
    answer = index.query(prompt, llm=chat_llama3)

    print("Llama3 Chatbot: " + answer)
```

You can check the complete conversation about our dataset here

Enter your query: can you check job title for location in London?

Llama3 Chatbot: Here are the job titles with their corresponding locations

there are two job titles with their location in London:

1. Marketing Specialist: Finance, Small, London

2. Sales Manager: Retail, Medium, Sydney -> no, actually...

3. AI Researcher: Retail, Large, London

I hope that helps!

Enter your query: Can you check location of Large company size?

Llama3 Chatbot: Based on the data provided, here are the locations for companies with a large company size:

* Marketing Specialist: Large (Singapore)

* AI Researcher: Large (San Francisco and London)

Let me know if I can help with anything else!

Enter your query: can you list job title that need Python skills?

Llama3 Chatbot: Here are the job titles from the provided data that require Python skills:

- * Marketing Specialist
- * AI Researcher (multiple instances)
- * Software Engineer

Note: Some other roles may also require Python skills not listed here. This answer is based on the specific data provided.

Enter your query: can you check industry that has high AI adoption level?

Llama3 Chatbot: Based on the provided data, I see the following industries with High AI Adoption Level:

- * Entertainment (has multiple instances)
- * Finance (has multiple instances)

Enter your query: can you check job title that is not remote friendly ?

Llama3 Chatbot: Based on the provided data, I can identify the following job titles that are not "Remote Friendly":

1. Sales Manager (Retail, Small, Berlin)
2. UX Designer (Education, Large, San Francisco)
3. AI Researcher (Manufacturing, Large, Tokyo)

Enter your query: can you list location and industry for only software engineer?

Llama3 Chatbot: Based on the provided context, here is a list of locations and industries for Software Engineers:

1. Manufacturing
 - Singapore (Medium company size)
2. Entertainment
 - Dubai (Small company size)

Enter your query: can you mention job title and industry that has stable job growth projection?

Llama3 Chatbot: Based on the provided context, here are some examples of Job Title and Industry with a Stable job growth projection:

- Product Manager, Finance (Large, San Francisco)

- AI Researcher, Retail (Large, London)

As we see from the conversation we developed chatbot that can interact with external knowledge data source using RAG technique and Llama 3.1 8B large language model that run locally on our device.

You can find the dataset and source code for this RAG Ollama-Chatbot project on this [GitHub repository](#).