

# LLMs and Chat Models

## Instructor

Dipanjan Sarkar

Head of Community & Principal AI Scientist at Analytics Vidhya

Google Developer Expert - ML & Cloud Champion Innovator

Published Author

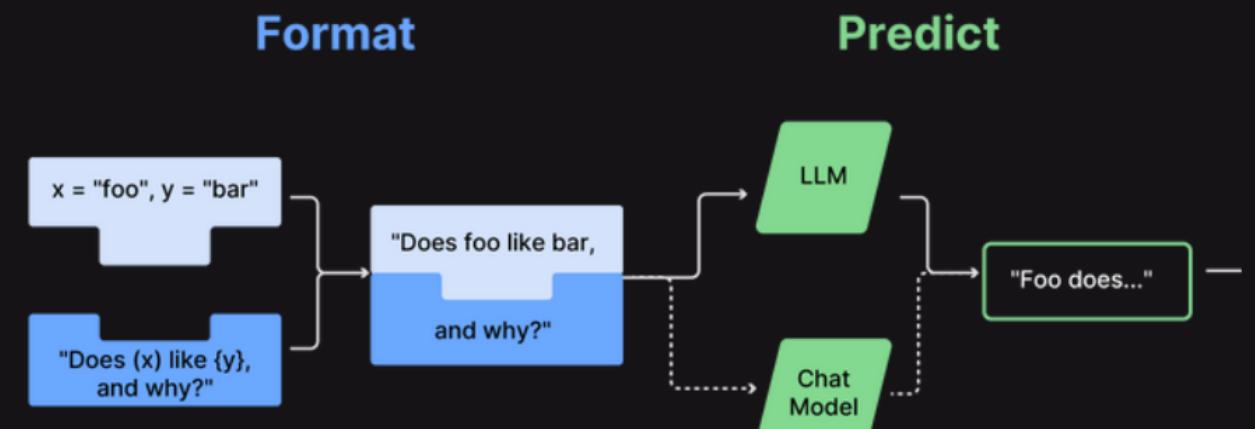


# Outline

- LLMs and Chat Models
- LangChain LLMs
- LangChain Chat Models
- LangChain integration with popular LLMs

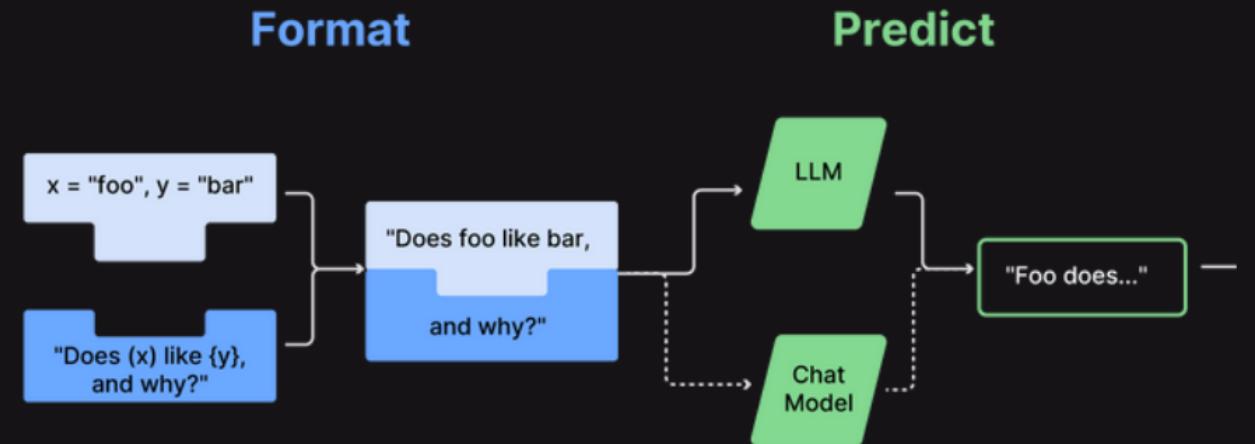
# LLMs and Chat Models

- LangChain provides two different ways of interfacing with LLMs
  - Regular LLMs for general prompting
  - Chat Models for conversational prompting
- We can interface with these LLMs using Prompt Templates in LangChain



# LangChain LLMs

- LLMs in LangChain refer to pure text completion models
- The APIs provided by LangChain take a string prompt as input and output a string completion text
- Typically meant for ad-hoc prompting and not for conversations



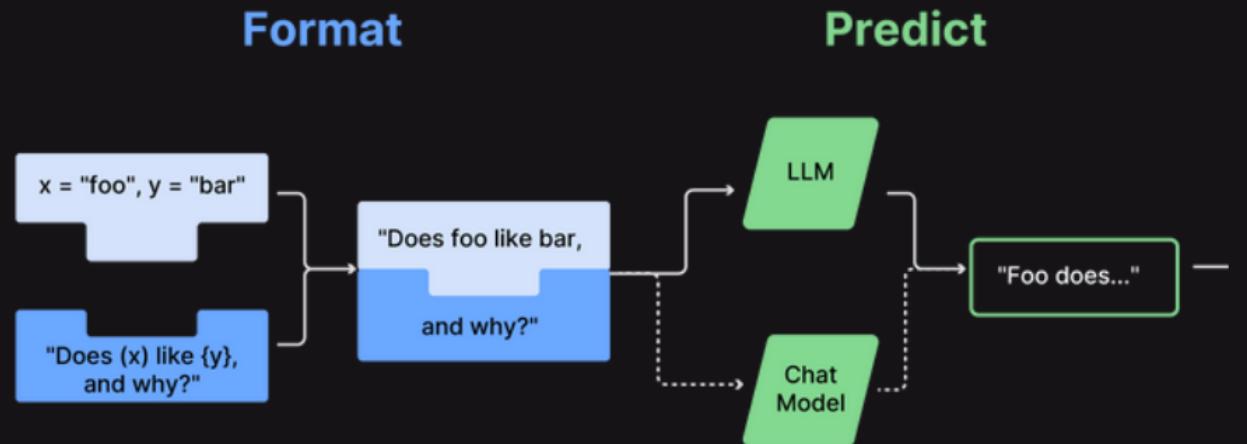
# LangChain LLMs

- LLM APIs typically connect to a specific model provider based on your choice of LLM
- Input is typically a string-based prompt
- Response is a string-based text output

```
● ● ●  
from langchain_openai import OpenAI  
  
chatgpt = OpenAI(model_name="gpt-3.5-turbo-instruct",  
                  temperature=0)  
  
prompt = """Explain what is Generative AI in 3 bullet points"""  
response = chatgpt.invoke(prompt)  
print(response)  
  
# Output  
# 1. Generative AI is a subset of a .....
```

# LangChain Chat Models

- Chat model APIs are often backed by LLMs but created specifically for having conversations
- They use a different interface than pure text completion models in the LLM API
- Instead of a single string, they take a list of chat messages as input and they return an AI message as output.



# LangChain Chat Models

- Chat model APIs can connect with any LLMs that support conversational prompting
- This API typically takes in a list of messages as the input prompt
- Output is usually an object of type AIMessage from which you can extract the response content which is a text output

```
● ● ●

from langchain_openai import ChatOpenAI
from langchain_core.messages import HumanMessage, SystemMessage

chatgpt = ChatOpenAI(model_name="gpt-3.5-turbo",
                     temperature=0)

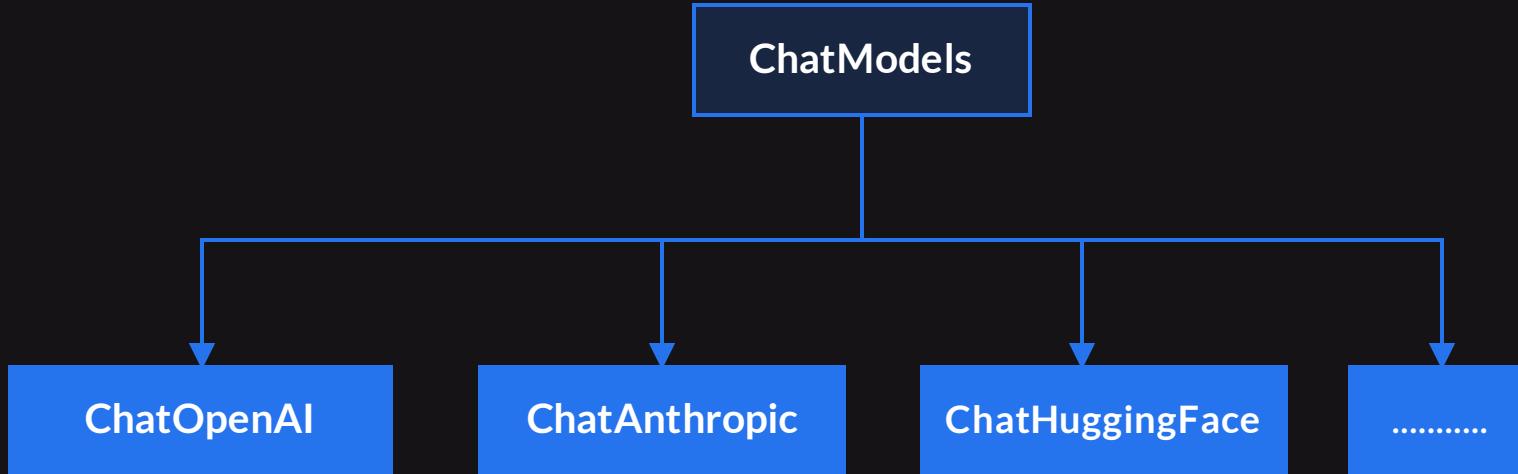
prompt = """Can you explain what is Generative AI in 3 bullet points?"""
sys_prompt = """Act as a helpful assistant
and give meaningful examples in your responses."""

messages = [
    SystemMessage(content=sys_prompt),
    HumanMessage(content=prompt),
]

response = chatgpt.invoke(messages)
print(response)

# Output
# AIMessage(content="1. Generative AI is a type of artificial ....")
```

# LangChain integration with popular LLMs



- LangChain enables you to have access to over 50+ popular LLMs which are accessible as Chat Models for conversational prompting

# Thank You

---