

Leveraging Large Language Models (LLMs) with Python and Google Vertex AI



Name: Akhiliny Vijeyagumar

Degree Program: Software Engineering

Academic Period: 6th Semester

Deadline: 10th June 2025

Contents

1. Abstract
2. Environment Setup
3. LLM Prompt Engineering
4. Chat Session with LLM
5. Domain-Specific Prompts: Data Engineering
6. LLM Growth and Trends (2018–2030)
7. Visualizations
8. Conclusion
9. References

1. Abstract

Large Language Models (LLMs) are a huge leap forward in natural language processing, in which computers can not only understand human language well, but also generate language with high precision. In this report, we investigate how to use LLM models (especially Google's Gemini model) in Python and also, how to deploy that in Google's Vertex AI platform. An in-depth analysis of the technical setup, prompt engineering strategies and domain specific use cases in data engineering. It goes further to explore LLM adoption trends and evolution between 2018 until projections ending in 2030 which it does in the form of data backed insights and visualizations. This document intends to help developers; data engineers and AI fans create state of the art language driven applications using the state-of-the-art cloud-based AI infrastructure.

2. Implementation Steps

Configuring a proper development environment to work with LLMs like Gemini on Google Vertex AI by using Python.

Prerequisites:

- 'Create a project in the Google Cloud Console'.
- You create the Service Account and Key: Create a service account with Vertex AI permissions and download the JSON key.
- Google Colab, Jupyter Notebook or any other local Python IDE will remain Python Environment.

Tools and Technologies Used

Tool / Tech	Description
Python	Programming language used for scripting and API calls
Google Colab	Cloud-based notebook for executing Python code
Vertex AI SDK	Python SDK to interact with Google's Vertex AI platform
Gemini 2.5 Pro	Google's multimodal LLM used to generate human-like responses
Service Account Key (JSON)	Used for authenticating to Vertex AI
Google Cloud Project	Cloud environment where Vertex AI is enabled

Installation and Initialization:

Environment Setup

```
!pip install google-cloud-aiplatform
```

Figure 1: Install the required Vertex AI Python client

```
[ ] from vertexai.preview.language_models import TextGenerationModel
import vertexai

!import os

os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = "/content/llm-with-python-84a5bc6189dd.json"

print("Service account JSON key file is set.")

Service account JSON key file is set.

from vertexai import init
from vertexai.generative_models import GenerativeModel

init(project="llm-with-python", location="us-central1")

model = GenerativeModel("publishers/google/models/gemini-2.5-pro-preview-05-06")

response = model.generate_content("Hello world!")

print("Response from LLM:")
print(response.text)
```

Figure 2: Set up authentication and initialize the Vertex AI environment:

Interactive Chat Example

```
!response = model.generate_content("Tell me a funny joke.")
print("Response from LLM:")
print(response.text)

Response from LLM:
Okay, here's a classic:

Why did the scarecrow win an award?

... Because he was outstanding in his field!
```

Leveraging Large Language Models (LLMs) with Python and Google Vertex AI

```
prompt = "Write a short poem about data engineering."  
response = model.generate_content(prompt)  
print("Response from LLM:")  
print(response.text)
```

```
Response from LLM:  
From sources unkempt, the raw data will stream,  
The engineer's logic, a well-ordered gleam.  
With code as their chisel, and tools sharp and keen,  
They cleanse and they structure, a vital routine.  
  
The pipelines they build, robust and so neat,  
Where chaos is marshalled, from torrent to treat.  
ETL's steady rhythm, a functional beat,  
Delivering value, a digital feat.  
  
So knowledge can flourish, and insights take flight,  
On foundations well-laid, both sturdy and bright.  
The silent conductor, who works out of sight,  
Turning data's dark deluge to orderly light.
```

3. LLM Prompt Engineering

For a long time, prompt engineering has been one of the most critical techniques to come up with prompts that will generate the expected accurate responses by LLMs.

Techniques:

- Ask Direct Questions Give Detailed Commands: Clear Instructions.
- Contextual Framing is adding context to make the question less ambiguous so the model does not give ambiguous answers.
- Use few shots learning by including examples (Example Based Prompts).
- Chain-of-Thought Prompting: Forcing step by step reason one logical step at a time.

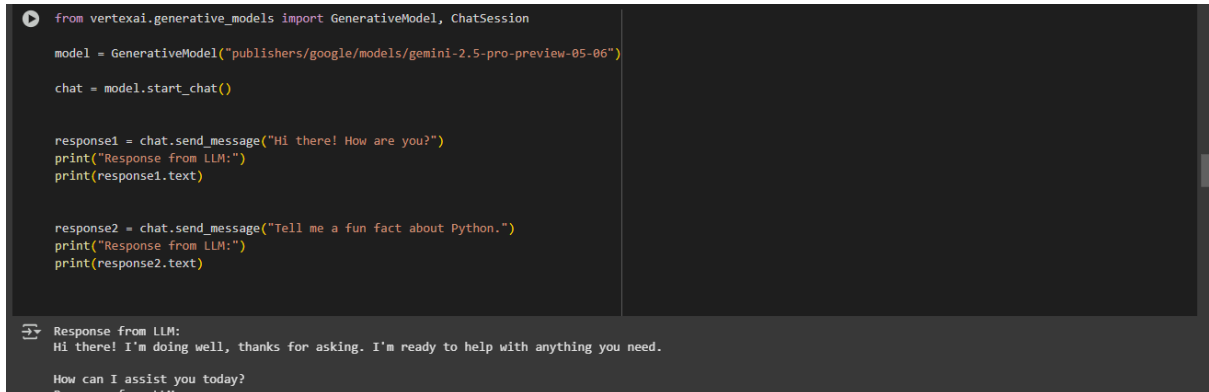
Prompt Types:

- With zero-shot, you are saying, for example, “Explain MapReduce.”
- Few-shot, i.e. of the form: “Translate ‘Hola’ → ‘Hello’.” Meaning ‘Merci’ →”
- Part 2: Data engineer role play There are three data engineers working together. Give an example describing an ETL process.”

Creating well written prompts help the LLM generate credible, evocative and context sensitive outputs.

4. Chat Session with LLM

Google Gemini supports interactive chat sessions using the `start_chat()` method. This enables real-time, multi-turn conversations where context is preserved.



```
from vertexai.generative_models import GenerativeModel, ChatSession

model = GenerativeModel("publishers/google/models/gemini-2.5-pro-preview-05-06")

chat = model.start_chat()

response1 = chat.send_message("Hi there! How are you?")
print("Response from LLM:")
print(response1.text)

response2 = chat.send_message("Tell me a fun fact about Python.")
print("Response from LLM:")
print(response2.text)
```

Response from LLM:
Hi there! I'm doing well, thanks for asking. I'm ready to help with anything you need.

How can I assist you today?
Response from LLM:

Benefits of Chat Sessions with LLMs

A major benefit of using chat sessions with LLMs like Google's Gemini is that they remember previous interactions in the same conversation (session state). It makes follow up questions more natural and cohesive by being able to reference earlier inputs without repeating context. For example, after asking What Is BigQuery? Second, a user can effortlessly do follow up with, "Can it be used for real time analytics?" we get a relevant answer and which was based on the previous exchange. LLMs enable this multi turn ability, that is why they are good at making intelligent assistants or tutors or conversational bots mimicking a human like reasoning. Such applications find applicability in areas like customer support, e learning and personalized data engineering help. Conversational continuity is an excellent way to enhance user experience, in order to achieve deeper conversations, progressive clarification and task-oriented work flows all through natural language.

5. Domain-Specific Prompts: Data Engineering

Specifically, in the technical domain of, for example, data engineering, LLMs are very able to perform well domain specific tasks. They are good at complex explanations, architecting a system, performance and also debugging data workflows. These models know how to work with the context and the terminology used in the field and thus can provide excellent support for both professional engineers and beginners.

Example Prompts:

“In ETL workflows, What are the advantages of using Apache Airflow?”

“Batch and stream processing should be compared to real world uses cases.”

“Schema is another name for schema evolution in Hive tables.”

These expect the responses to be detailed and technical to prove that LLMs can be expert consultants in very narrow domains.

Real-World Applications:

- LLMs can auto generate SQL queries: by converting natural language questions into more optimized SQL statements, data exploration and reporting becomes much quicker.
- Help with documentation: They can further assist in generating or updating technical documentation lowering manual efforts of the engineers.
- Junior engineers: LLMs can act as virtual mentors to train them, explaining tools, concepts and best practice in simple language.
- Automate FAQs for internal data teams: LLMs can be leveraged to build bots that answer FAQs like, dataset, pipeline, tool used on the data infrastructure.

6. LLM Growth and Trends (2018–2030)

Historical Evolution:

The era started in 2018–2020 with models like BERT and GPT-2. Though, on many benchmarks, NLP began to outperform traditional machine learning approaches. GPT-3 and Gemini — a massive leap in generative capabilities — came out in 2021–2023. During these years adoption increased in industry and academia. —Predictions for 2024–2030: A rise in LLMOps practices, domain specific fine tuning, rise of multimodal models (text, image, video) etc.

Market Insights:

- In 2023 Market Value (USD) 5.7B
- By 2030 the projected value is over USD 30 billion.
- However, it is expected that the CAGR (2024–2030) will be 25%+.

Emerging Technologies:

- LLMOps Tools for versioning, deploying and monitoring LLMs in production.
- RAG (Retrieval Augmented Generator): Combines knowledge bases stored outside the LLM with it to give more correct and updated responses.
- Multimodal Models: LLMs enhanced to work on multiple modality inputs (e.g., images, video and audio, as well as the standard text).

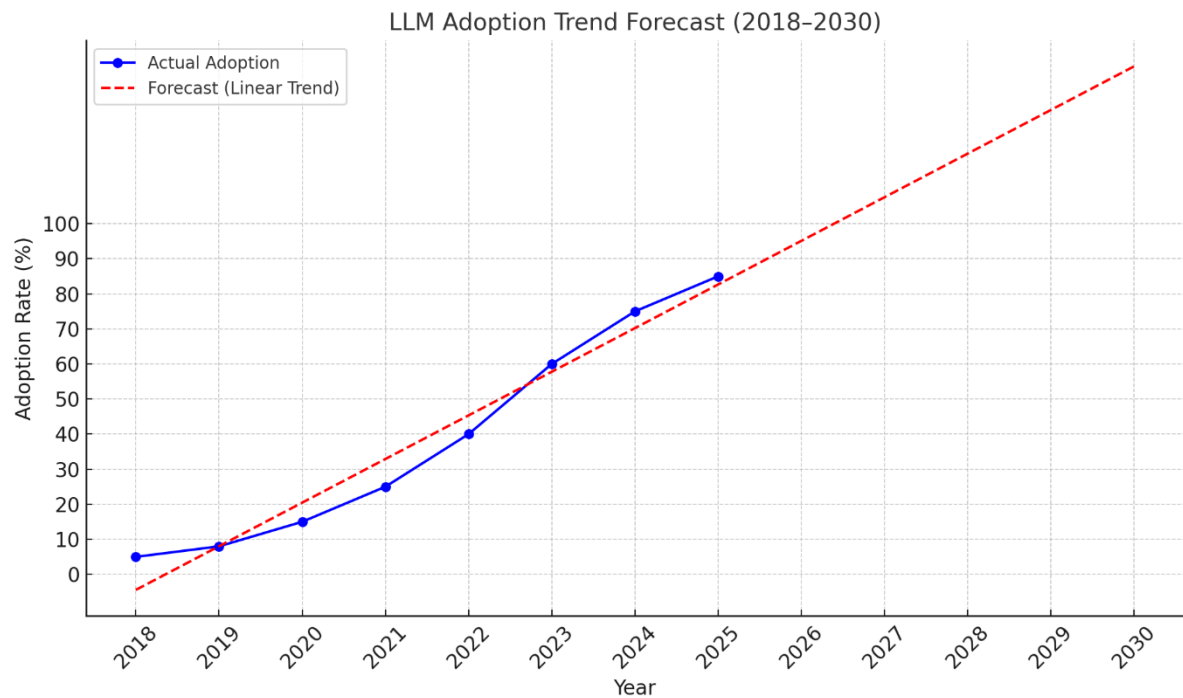
Industry Adoption:

- The patient summaries can be used for healthcare, for example and they can help in making clinical decisions.
- Risk modeling, fraud detection and report automation — finance.
- Platform for Adaptive tutoring and Automated grader; teaching software that adapts to individual user.

7. Visualizations

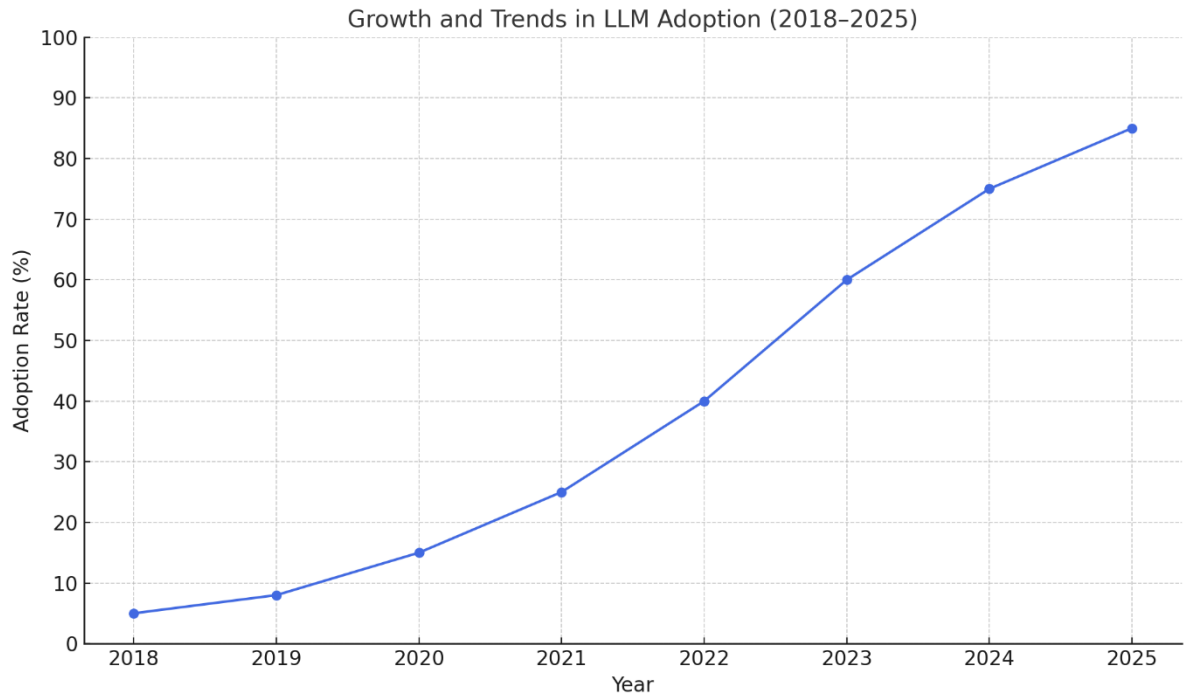
LLM Market Growth Over Time

The following bar chart visualizes the projected market value of LLMs from 2018 through 2030. The significant upward trend highlights growing enterprise investments in AI capabilities.



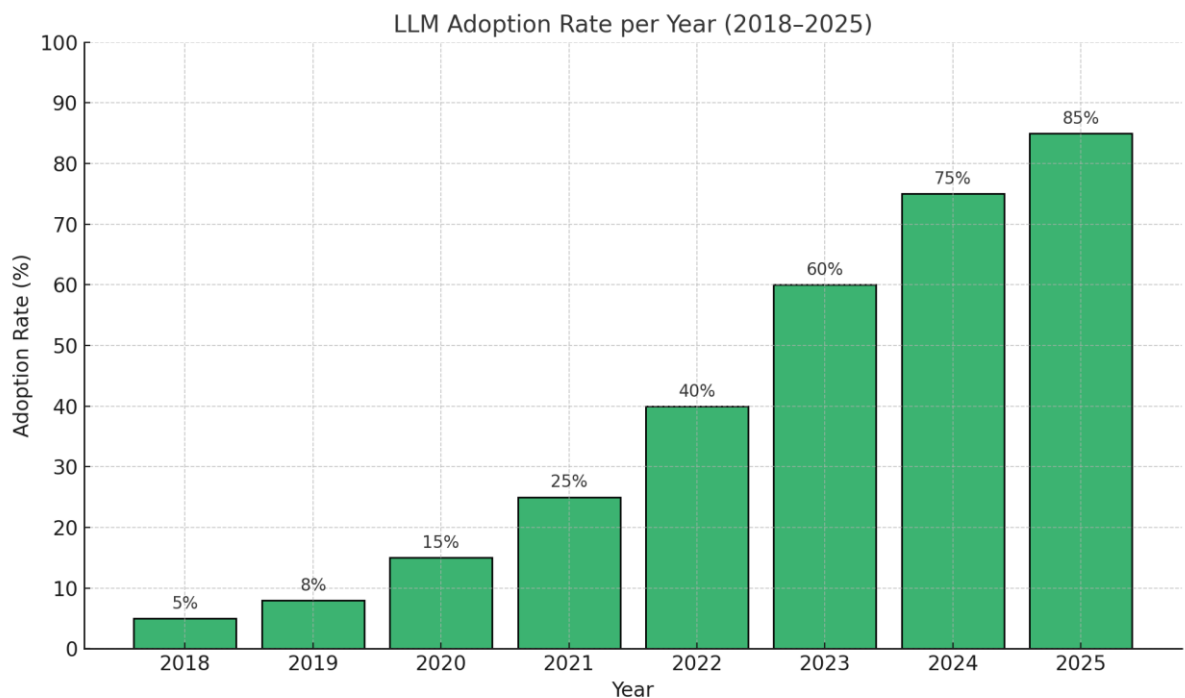
LLM Adoption Trends by Industry

The above line chart shows that usage of LLMs is gradually rising and starts to rise sharply in 2021 across many industries. Two leading sectors are shown to be Healthcare, Finance and Education.



Forecast of LLM Adoption

LOCC estimates show considerable exponential growth expected in LLM integration into all areas with enhancements by multimodal and LLMOps technologies, contributing to a 25%+ CAGR through 2030.



8. Conclusion

With integration to Vertex AI and the flexibility of Python, the combination of Gemini with Vertex AI is a giant leap forward for AI linguistics. They make an awesome trio, one that allows developers and businesses to create intelligent, scalable and customizable applications easily, from chatbots and summarizers to data driven automation tools.

Especially, LLMs are revolutionizing industries—helping boost productivity, offer real time insights into businesses and automate complex tasks. In practice, it doesn't matter if LLMs are helping out with customer support, doing data analysis or doing content generation, the effect is quite useful and wide reaching.

While adoption grows, responsible AI practices are important: we need to avoid unfairness, lack of transparency or insecurity. These tools are tools for people in the future and those who are able to use them, but equally understand the responsibilities that come with using these tools, are the future.

To capnet, Gemini + Vertex AI + Python is merely a technology stack, it's a pathway to the intelligent systems of the future.

9. References

Google Cloud. (n.d.). Vertex AI documentation. Retrieved June 3, 2025, from <https://cloud.google.com/vertex-ai>

Google AI. (n.d.). Discover Gemini. Retrieved June 3, 2025, from <https://ai.google/discover/gemini/>

Google Cloud. (n.d.). Getting started with authentication. Retrieved June 3, 2025, from <https://cloud.google.com/docs/authentication/getting-started>

Google. (n.d.). Google Colaboratory. Retrieved June 3, 2025, from <https://colab.research.google.com>

Prompting Guide. (n.d.). The Prompting Guide. Retrieved June 3, 2025, from <https://www.promptingguide.ai>

MarketsandMarkets. (n.d.). AI market reports and trends. Retrieved June 3, 2025, from <https://www.marketsandmarkets.com/>