What is Prompt Engineering

- **Definition**: Prompt engineering involves designing and refining input prompts to optimize the performance of AI models, particularly in natural language processing (NLP).
- **Purpose**: The goal is to guide AI models to produce accurate, relevant, and contextually appropriate outputs.
- **Relevance**: It is crucial for improving the interaction between humans and AI, enhancing the model's ability to understand and generate language.
- **Applications**: Used in chatbots, virtual assistants, content generation, and more.
- **Significance**: Plays a vital role in leveraging AI capabilities for business and technological advancements.

Prompt Engineering – Key

- **Prompt Design**: Crafting effective prompts that clearly convey the desired task or question to the AI model.
- **Contextual Understanding**: Ensuring the AI model comprehends the context of the prompt to generate meaningful responses.
- **Iterative Refinement**: Continuously improving prompts based on model performance and feedback.
- **Prompt Templates**: Using standardized templates to maintain consistency and efficiency in prompt creation.
- **Evaluation Metrics**: Assessing the effectiveness of prompts using metrics like accuracy, relevance, and coherence.

Prompt Engineering – Key

Prompt Design: Crafting effective prompts that clearly convey the desired task or question to the AI model.

Contextual Understanding: Ensuring the AI model comprehends the context of the prompt to generate meaningful responses.

Iterative Refinement: Continuously improving prompts based on model performance and feedback.

Prompt Templates: Using standardized templates to maintain consistency and efficiency in prompt creation.

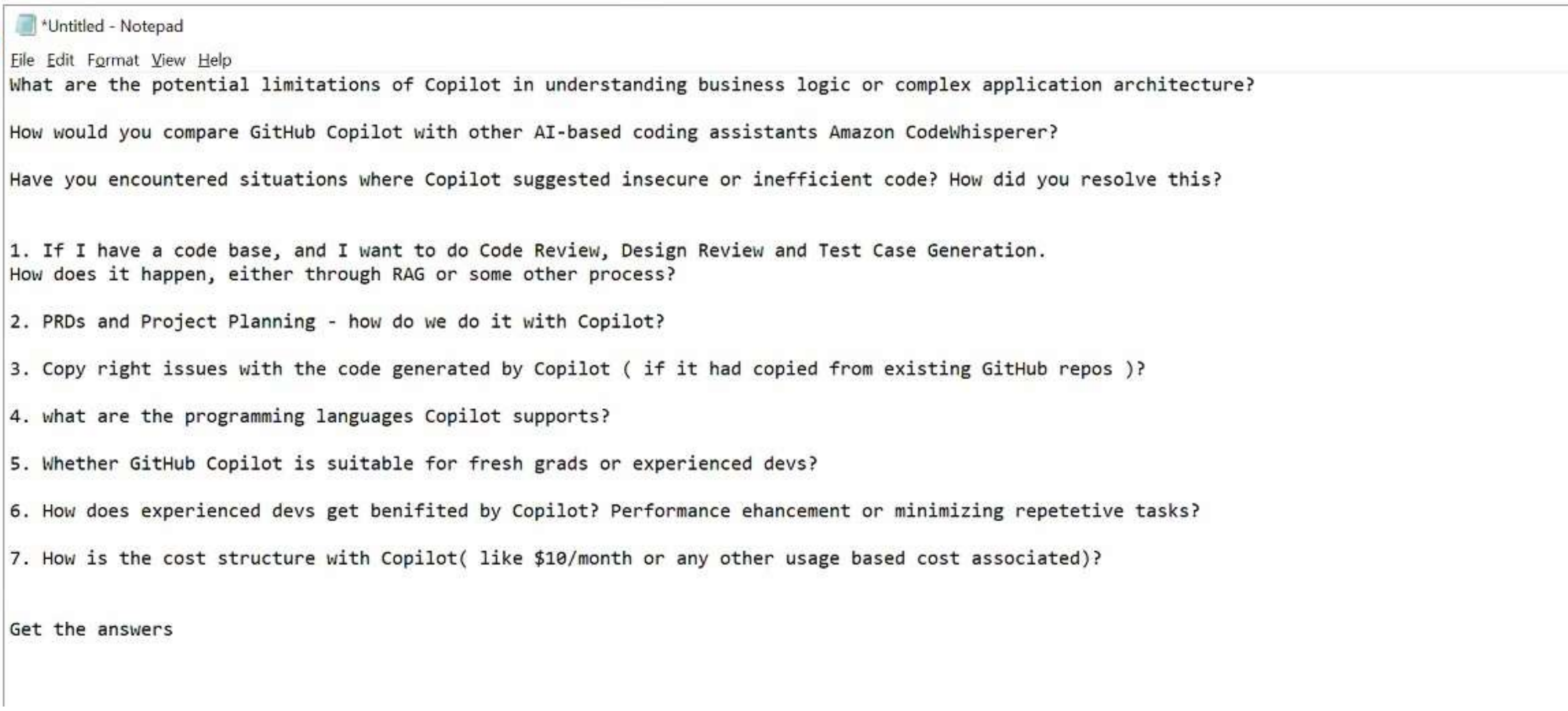
Evaluation Metrics: Assessing the effectiveness of prompts using metrics like accuracy, relevance, and coherence. →

{ system: I am a financial assistance. Get your rep
in 200 words max.

User: "Write an essay in 200 words"

Assistant: example 1: ______
example2: ______

}



What are the potential limitations of Copilot in understanding business logic or complex application architecture?

How would you compare GitHub Copilot with other AI-based coding assistants Amazon CodeWhisperer?

Have you encountered situations where Copilot suggested insecure or inefficient code? How did you resolve this?

1. If I have a code base, and I want to do Code Review, Design Review and Test Case Generation.
How does it happen, either through RAG or some other process?

2. PRDs and Project Planning - how do we do it with Copilot?

3. Copy right issues with the code generated by Copilot (if it had copied from existing GitHub repos)?

4. what are the programming languages Copilot supports?

5. Whether GitHub Copilot is suitable for fresh grads or experienced devs?

6. How does experienced devs get benifited by Copilot? Performance ehancement or minimizing repetetive tasks?

7. How is the cost structure with Copilot(like $10/month or any other usage based cost associated)?

Get the answers

[redacted]

Prompt Engineering – Key Techniques

1 v.

- **Few-Shot Learning**: Providing the model with a few examples within the prompt to improve its understanding and output.
- **Zero-Shot Learning**: Designing prompts that enable the model to perform tasks without prior examples, relying solely on its training.
- **Prompt Tuning**: Adjusting prompts to fine-tune model responses for specific tasks or domains.
- **Prompt Chaining**: Using a series of interconnected prompts to guide the model through complex tasks or multi-step processes.
- **Bias Mitigation**: Crafting prompts that minimize bias and ensure fair and balanced outputs from the AI model.

Career Options in Prompt Engineering

Emerging Field: Prompt engineering is a rapidly growing area within AI, offering numerous career opportunities.

Diverse Roles: Opportunities span various industries, including technology, healthcare, finance, and more.

Interdisciplinary Skills: Combines skills in AI, NLP, data science, and software development.

Innovation-Driven: Ideal for those interested in cutting-edge technology and creative problem-solving.

Prompt Engineer

- **Role Overview**: Specializes in designing and refining prompts to optimize AI model outputs.
- **Key Responsibilities**:
  - Crafting and testing effective prompts for AI models.
  - Analyzing model performance and iterating on prompt designs.
  - Collaborating with data scientists and AI researchers.
  - Implementing prompt engineering techniques to improve accuracy.
  - Documenting prompt strategies and outcomes.
  - Staying updated with the latest AI advancements.
- **Skills Required**: Strong understanding of NLP, AI model behavior, and data analysis.

[redacted]

AI/NLP Specialist

- **Role Overview**: Develops and implements AI and NLP solutions across various applications.
- **Key Responsibilities**:
  - Designing and training NLP models for specific tasks.
  - Integrating AI solutions into existing systems.
  - Conducting research to improve NLP algorithms.
  - Collaborating with cross-functional teams to deploy AI solutions.
  - Evaluating and optimizing model performance.
  - Writing technical documentation and reports.
- **Skills Required**: Proficiency in machine learning, programming languages (Python, R), and NLP frameworks.

[redacted]

Chatbot Developer

- **Role Overview**: Builds and maintains conversational AI systems, such as chatbots and virtual assistants.
- **Key Responsibilities**:
  - Designing dialogue flows and user interactions.
  - Implementing AI models for natural language understanding.
  - Testing and refining chatbot responses.
  - Integrating chatbots with various platforms and APIs.
  - Monitoring chatbot performance and user feedback.
  - Collaborating with UX designers to enhance user experience.
- **Skills Required**: Experience with chatbot platforms, conversational design, and user experience (UX) principles.

AI Product Manager

- **Role Overview**: Manages the development and deployment of AI-powered products and solutions.
- **Key Responsibilities**:
  - Defining product vision and strategy.
  - Coordinating cross-functional teams for product development.
  - Ensuring alignment with business goals and user needs.
  - Overseeing product lifecycle from concept to launch.
  - Analyzing market trends and competitor products.
  - Managing product budgets and timelines.
- **Skills Required**: Strong project management, communication, and technical understanding of AI technologies.

AI Research Scientist

- **Role Overview**: Conducts research to advance the field of AI, focusing on new algorithms and methodologies.
- **Key Responsibilities**:
  - Designing and conducting experiments to test AI theories.
  - Publishing research papers and presenting findings at conferences.
  - Collaborating with academic and industry partners.
  - Developing innovative AI models and techniques.
  - Mentoring junior researchers and interns.
  - Securing funding for research projects.
- **Skills Required**: Deep knowledge of AI theory, statistical analysis, and research methodologies.

Job Opportunities in the United States

- **High Demand for AI Specialists**: The US has a robust demand for AI specialists, including prompt engineers and NLP experts.
- **Tech Hubs**: Cities like San Francisco, New York, and Seattle are leading in AI job openings.
- **Diverse Industries**: Opportunities span tech, healthcare, finance, and retail sectors.
- **Remote Work Options**: Many companies offer remote positions, expanding access to roles nationwide.
- **Competitive Salaries**: The US offers some of the highest salaries for AI roles globally.
- **Innovation and Research**: Strong focus on AI research and development, with numerous startups and established companies investing in AI.

Job Opportunities in India

- **Tech Industry Growth**: India is a major player in the global tech industry, with a growing focus on AI.
- **IT Hubs**: Bangalore, Hyderabad, and Pune are leading cities for AI job opportunities.
- **Startup Ecosystem**: Vibrant startup scene with numerous AI-focused companies.
- **Cost-Effective Talent**: Competitive salaries and a large pool of skilled professionals.
- **Educational Initiatives**: Increasing emphasis on AI education and training programs.
- **Global Collaboration**: Opportunities to work with international clients and projects.

[redacted]

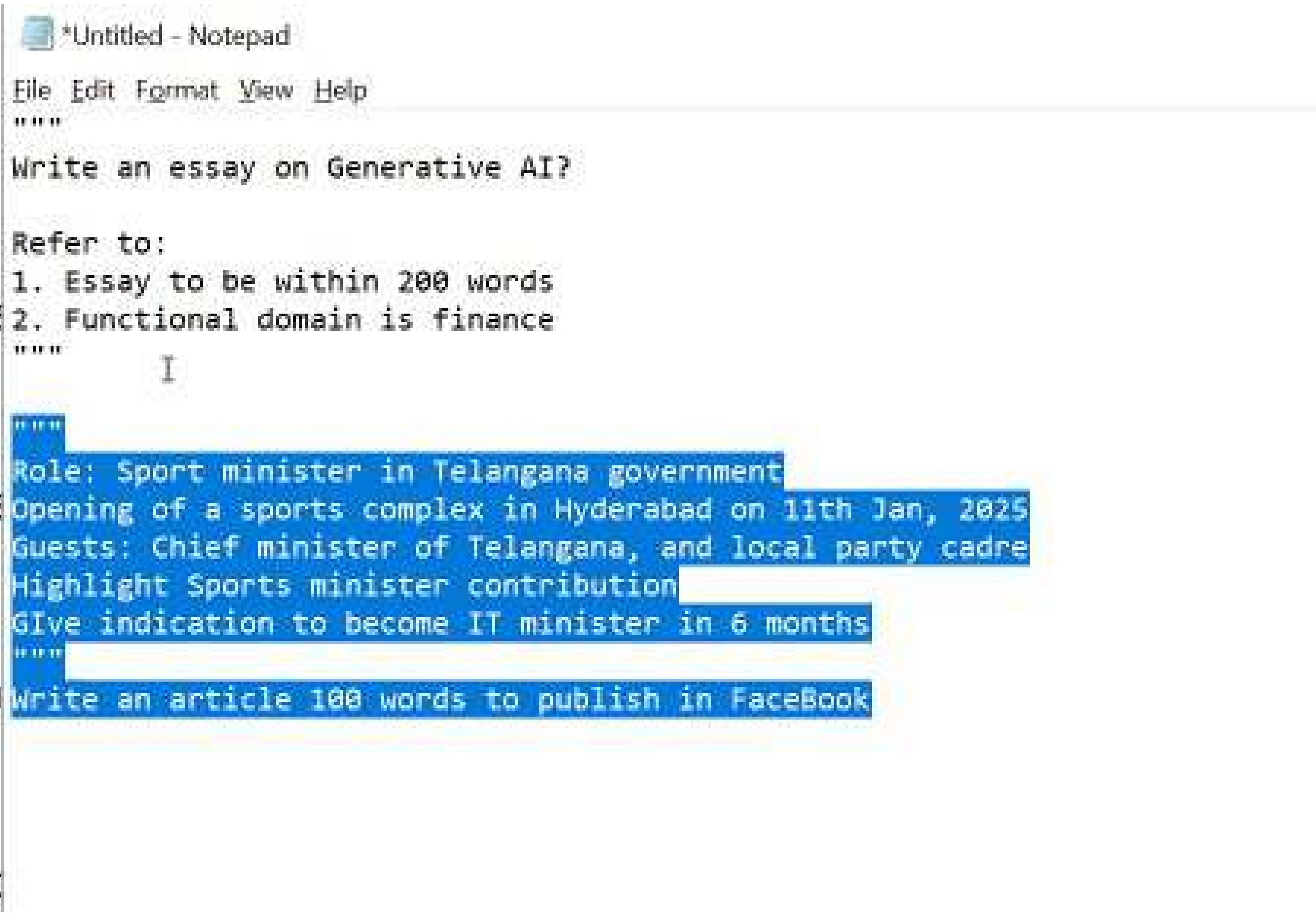
Emerging Markets and Future Trends

- **AI Expansion in Africa**: Growing interest in AI technologies, with opportunities in tech hubs like Nairobi and Lagos.
- **Latin America Growth**: Countries like Brazil and Mexico are investing in AI, creating new job opportunities.
- **Asia-Pacific Developments**: Rapid AI adoption in countries like China, Japan, and South Korea.
- **Focus on Sustainability**: AI roles in projects aimed at environmental sustainability and renewable energy.
- **Cross-Industry Applications**: Increasing demand for AI skills in non-traditional sectors like agriculture and logistics.
- **Future-Proof Careers**: AI roles offer long-term career stability and growth potential.

[redacted]

Daily Responsibilities of a Prompt Engineer

- **Designing Effective Prompts**: Craft and refine prompts to optimize AI model outputs, ensuring clarity and relevance.
- **Collaborating with Teams**: Work closely with data scientists, AI researchers, and product managers to align prompt strategies with project goals.
- **Analyzing Model Performance**: Evaluate AI model responses to prompts, identifying areas for improvement and iterating on designs.
- **Research and Development**: Stay updated with the latest advancements in AI and NLP to incorporate new techniques and methodologies.
- **Documentation and Reporting**: Maintain detailed records of prompt designs, testing outcomes, and performance metrics for future reference.
- **Problem-Solving**: Address challenges related to model behavior, prompt ambiguity, and unexpected outputs, developing creative solutions.



```
"""
Write an essay on Generative AI?

Refer to:
1. Essay to be within 200 words
2. Functional domain is finance
"""

"""
Role: Sport minister in Telangana government
Opening of a sports complex in Hyderabad on 11th Jan, 2025
Guests: Chief minister of Telangana, and local party cadre
Highlight Sports minister contribution
GIve indication to become IT minister in 6 months
"""
Write an article 100 words to publish in FaceBook
```

[redacted]

Work Environment and Career Development

- **Dynamic Work Environment**: Engage in a fast-paced, innovative setting where adaptability and creativity are key.
- **Continuous Learning**: Participate in workshops, conferences, and online courses to enhance skills and stay current with industry trends.
- **Networking Opportunities**: Build connections with professionals in AI and related fields through industry events and collaborations.
- **Career Growth**: Opportunities for advancement into senior engineering roles, AI research positions, or leadership roles in AI strategy.
- **Cross-Functional Projects**: Involvement in diverse projects across various industries, from tech and finance to healthcare and entertainment.
- **Work-Life Balance**: Many roles offer flexible work arrangements, including remote work options, to support a healthy work-life balance.

[redacted]

Career Advancement Opportunities for Prompt Engineers

- **Transition to Senior Roles**: With experience, prompt engineers can advance to senior positions, taking on more complex projects and leadership responsibilities.
- **Specialization in AI Domains**: Opportunities to specialize in areas like NLP, machine learning, or AI ethics, enhancing expertise and career prospects.
- **Cross-Functional Leadership**: Potential to lead cross-functional teams, collaborating with data scientists, developers, and product managers to drive AI initiatives.
- **AI Research and Development**: Engage in cutting-edge research, contributing to the development of new AI models and technologies.
- **Product Management**: Transition into product management roles, overseeing the development and deployment of AI-driven products and solutions.
- **Consulting and Advisory Roles**: Provide expert guidance to organizations on AI strategy and implementation, leveraging prompt engineering skills.

[redacted]

Industry Trends and Long-Term Growth Potential

Expanding AI Applications: As AI applications continue to grow across industries, demand for skilled prompt engineers is expected to rise.

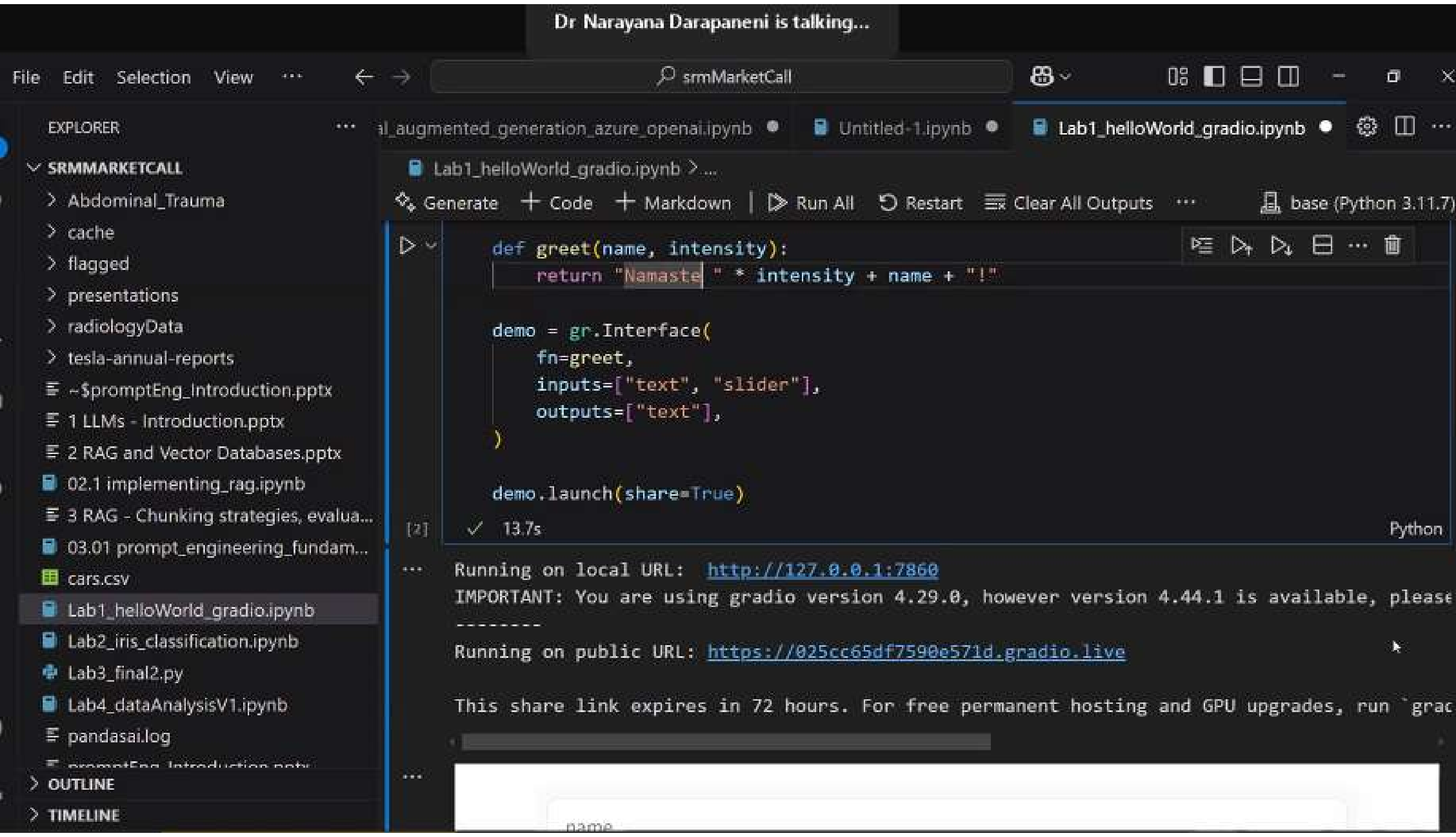
Integration with Emerging Technologies: Opportunities to work with emerging technologies like augmented reality, IoT, and blockchain, broadening career horizons.

Global Opportunities: Increasing demand for AI expertise worldwide, offering opportunities for international roles and collaborations.

Continuous Learning and Upskilling: Importance of staying updated with the latest AI advancements and continuously enhancing skills to remain competitive.

Entrepreneurial Ventures: Potential to start AI-focused businesses or consultancies, leveraging expertise in prompt engineering and AI technologies.

Impact on Business and Society: Play a key role in shaping the future of AI, influencing how businesses operate and how society interacts with technology.


Dr Narayana Darapaneni is talking...
File
Edit
Selection
View
srmMarketCall
EXPLORER
SRMMARKETCALL
Abdominal_Trauma
cache
flagged
presentations
radiologyData
tesla-annual-reports
~$promptEng_Introduction.pptx
1 LLMs - Introduction.pptx
2 RAG and Vector Databases.pptx
02.1 implementing_rag.ipynb
3 RAG - Chunking strategies, evalua...
03.01 prompt_engineering_fundam...
cars.csv
Lab1_helloWorld_gradio.ipynb
Lab2_iris_classification.ipynb
Lab3_final2.py
Lab4_dataAnalysisV1.ipynb
pandasai.log
OUTLINE
TIMELINE
al_augmented_generation_azure_openai.ipynb
Untitled-1.ipynb
Lab1_helloWorld_gradio.ipynb
Lab1_helloWorld_gradio.ipynb > ...
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
base (Python 3.11.7)
[2]
13.7s
Python
name

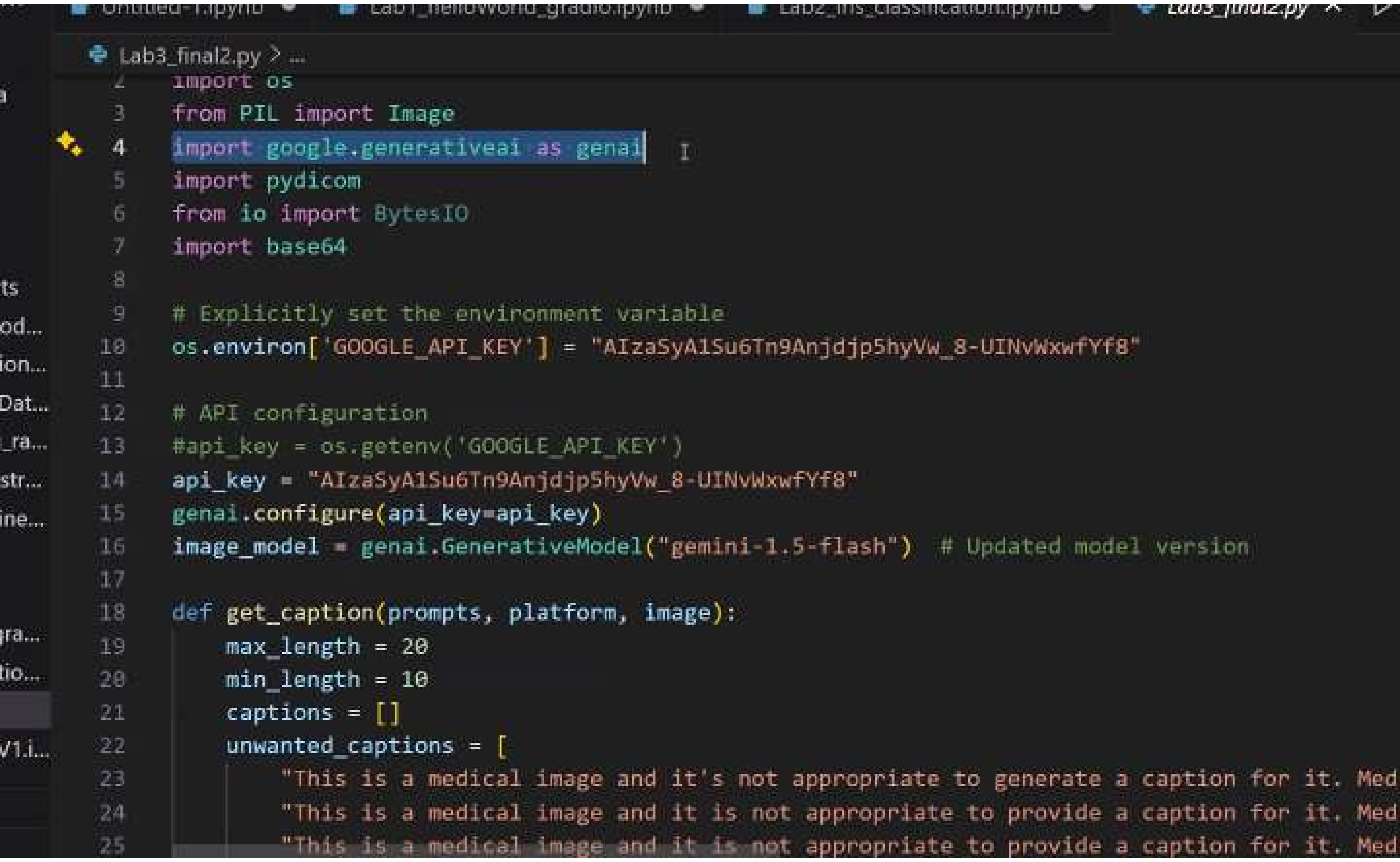
```
def greet(name, intensity):
    return "Namaste " * intensity + name + "!"

demo = gr.Interface(
    fn=greet,
    inputs=["text", "slider"],
    outputs=["text"],
)

demo.launch(share=True)
```

```
Running on local URL:  http://127.0.0.1:7860
IMPORTANT: You are using gradio version 4.29.0, however version 4.44.1 is available, please
--------
Running on public URL: https://025cc65df7590e571d.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `grac
```



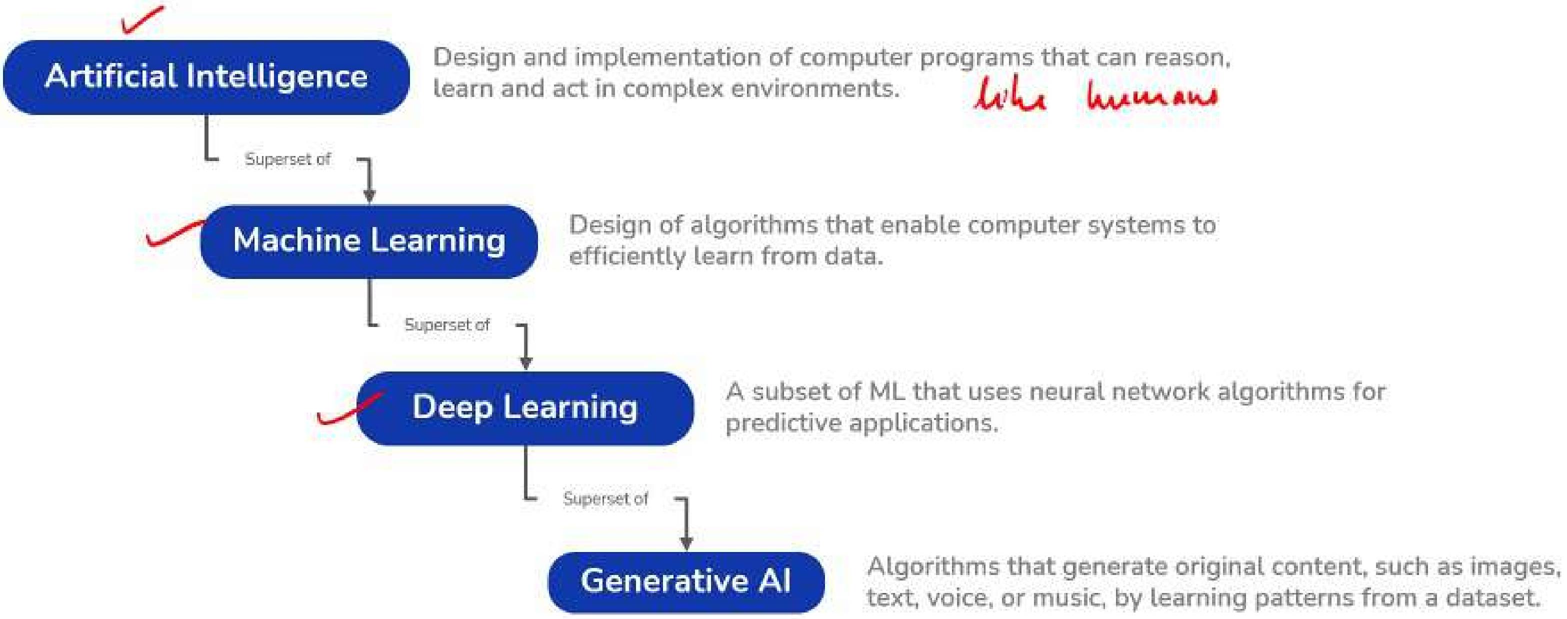
```
import os
from PIL import Image
import google.generativeai as genai
import pydicom
from io import BytesIO
import base64

# Explicitly set the environment variable
os.environ['GOOGLE_API_KEY'] = "AIzaSyA1Su6Tn9Anjdjp5hyVw_8-UINvWxwfYf8"

# API configuration
#api_key = os.getenv('GOOGLE_API_KEY')
api_key = "AIzaSyA1Su6Tn9Anjdjp5hyVw_8-UINvWxwfYf8"
genai.configure(api_key=api_key)
image_model = genai.GenerativeModel("gemini-1.5-flash")  # Updated model version

def get_caption(prompts, platform, image):
    max_length = 20
    min_length = 10
    captions = []
    unwanted_captions = [
        "This is a medical image and it's not appropriate to generate a caption for it. Med
        "This is a medical image and it is not appropriate to provide a caption for it. Med
        "This is a medical image and it is not appropriate to provide a caption for it. Med
```

Generative AI - An Introduction



Artificial Intelligence

Design and implementation of computer programs that can reason, learn and act in complex environments. like humans

Superset of

Machine Learning

Design of algorithms that enable computer systems to efficiently learn from data.

Superset of

Deep Learning

A subset of ML that uses neural network algorithms for predictive applications.

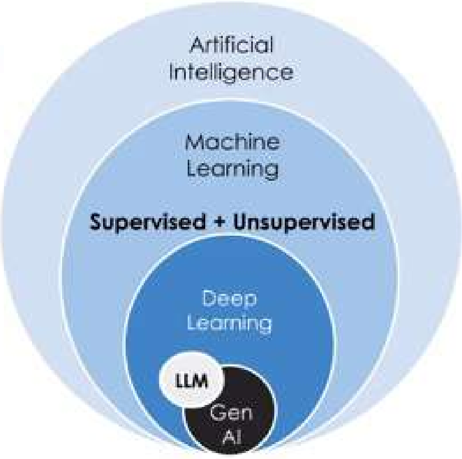
Superset of

Generative AI

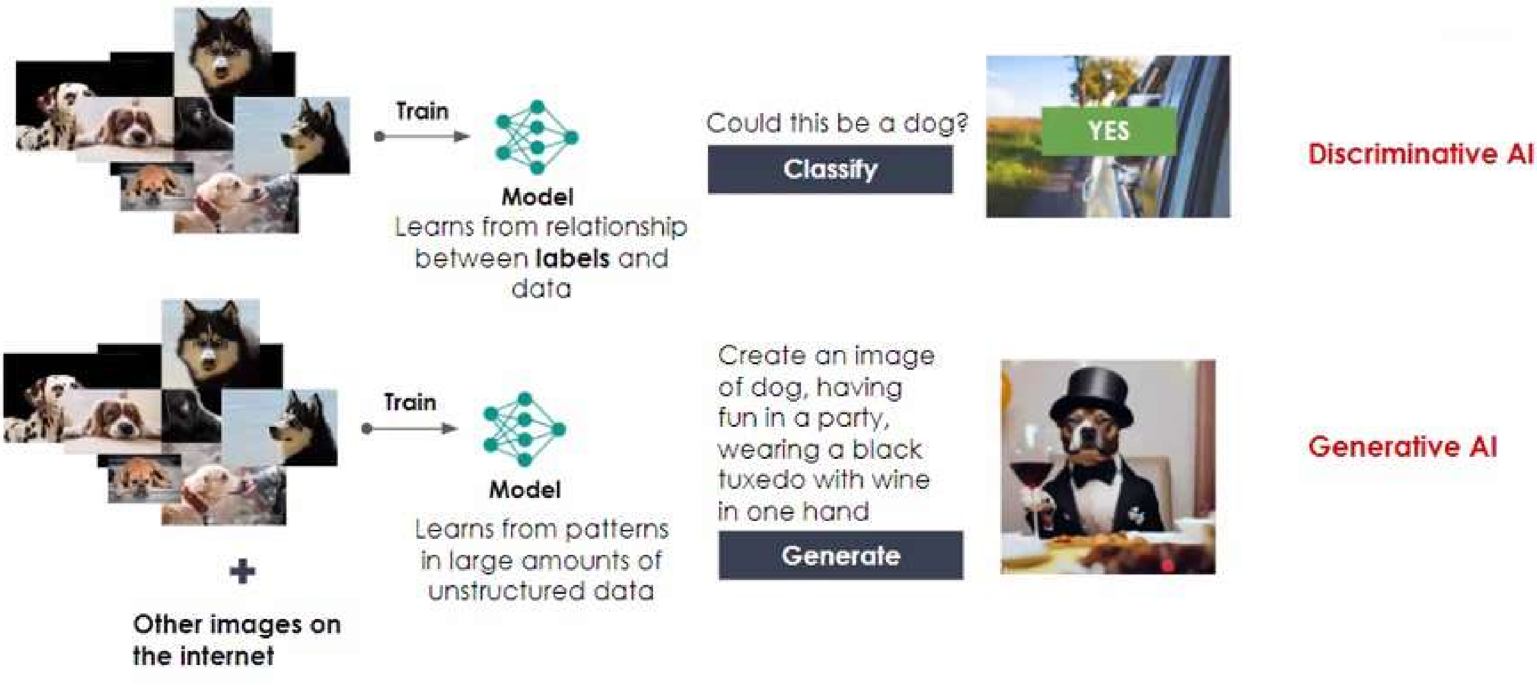
Algorithms that generate original content, such as images, text, voice, or music, by learning patterns from a dataset.

Getting Definitions Right

- 🖥️ **Artificial Intelligence (AI)** is the concept of computers simulating intelligence, achieved through algorithms that imitate human intelligence.
- 💻 **Machine Learning (ML)** involves computers learning rules from data, transitioning from traditional rule-based programming to data-driven learning.
- 🤖 **Deep Learning** is a subset of ML, specialized in processing unstructured data like images, text, audio, and video, using Neural Networks with extra layers for hierarchical data.
- 🎨 **Generative AI builds on Deep Learning** by allowing models to not only understand data but also generate novel content, such as text or images, based on learned probability distributions.
- 📜 **Large Language Models (LLMs)** serve as an interface between Deep Learning and Generative AI, enabling human communication with AI models through Natural Language prompts, offering conditional probability-based outputs tailored to user requests.



Discriminative AI vs Generative AI



Train

Model

Learns from relationship between **labels** and data

Could this be a dog?

Classify

Discriminative AI

Train

Model

Learns from patterns in large amounts of unstructured data

Create an image of dog, having fun in a party, wearing a black tuxedo with wine in one hand

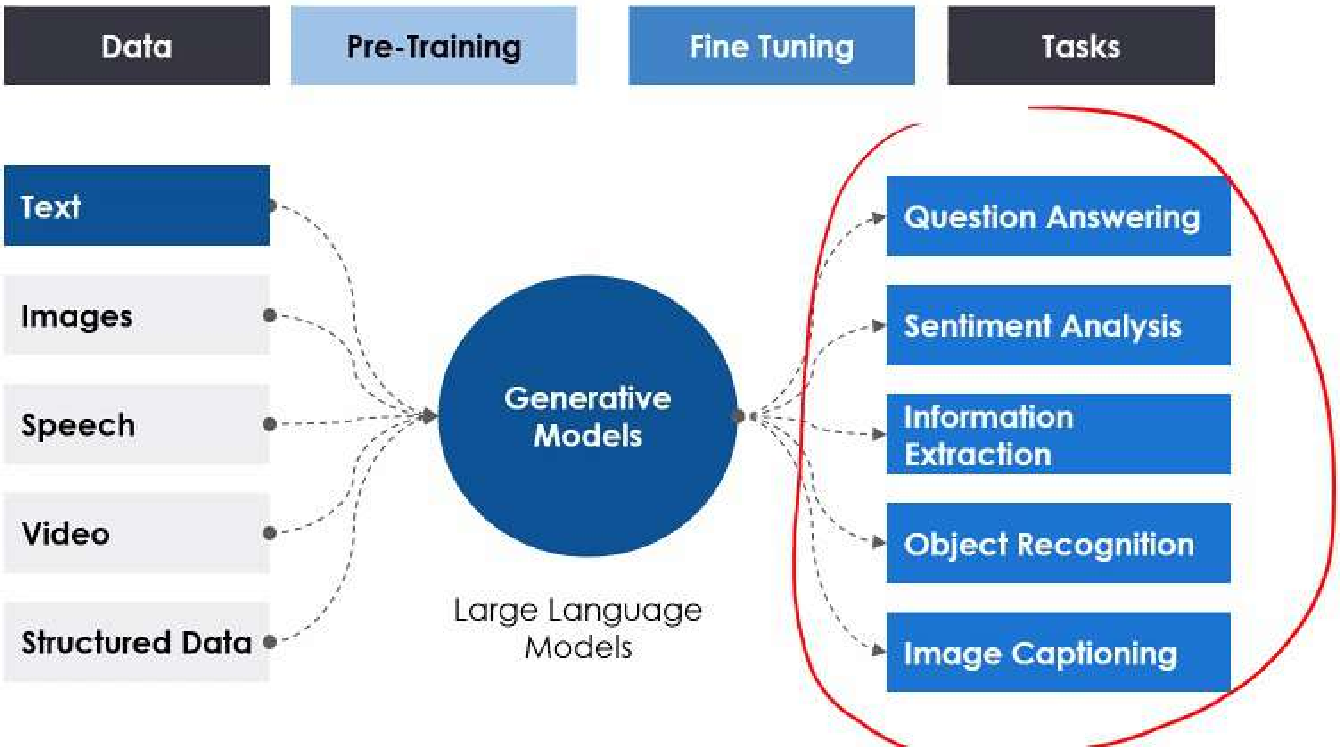
Generate

Generative AI

+

Other images on the internet

A Peek into Generative Models



Data

Pre-Training

Fine Tuning

Tasks

Text

Images

Speech

Video

Structured Data

Generative Models

Large Language Models

Question Answering

Sentiment Analysis

Information Extraction

Object Recognition

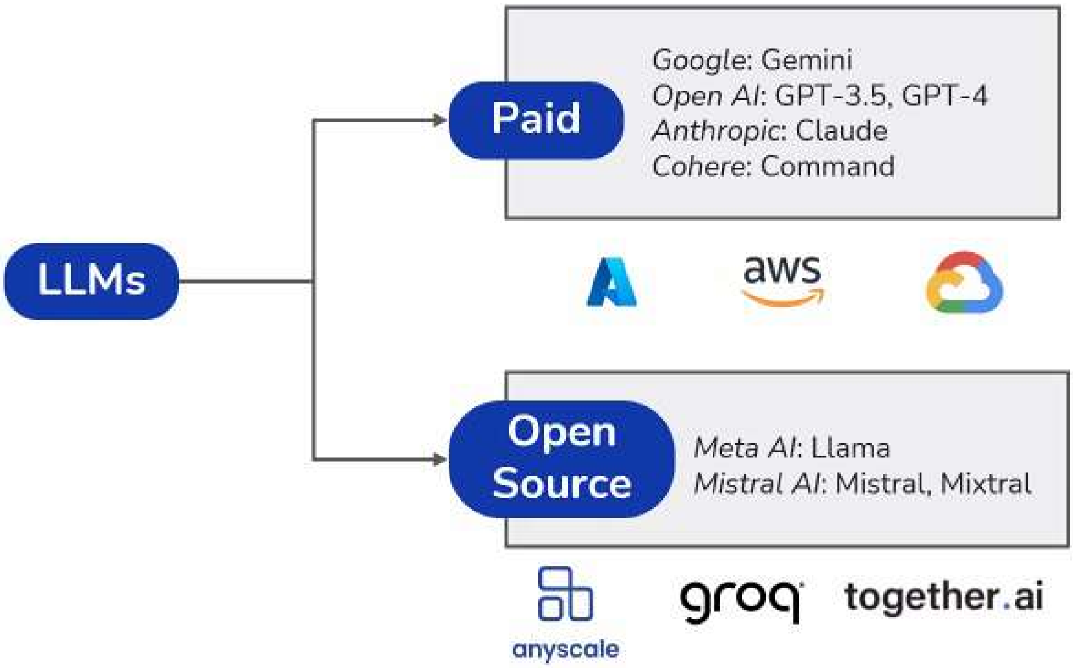
Image Captioning

LLM Applications - Shortcomings

- Vulnerability to injection attacks
- Data privacy concerns
- Intellectual property concerns on LLM outputs
- Lack of explainability

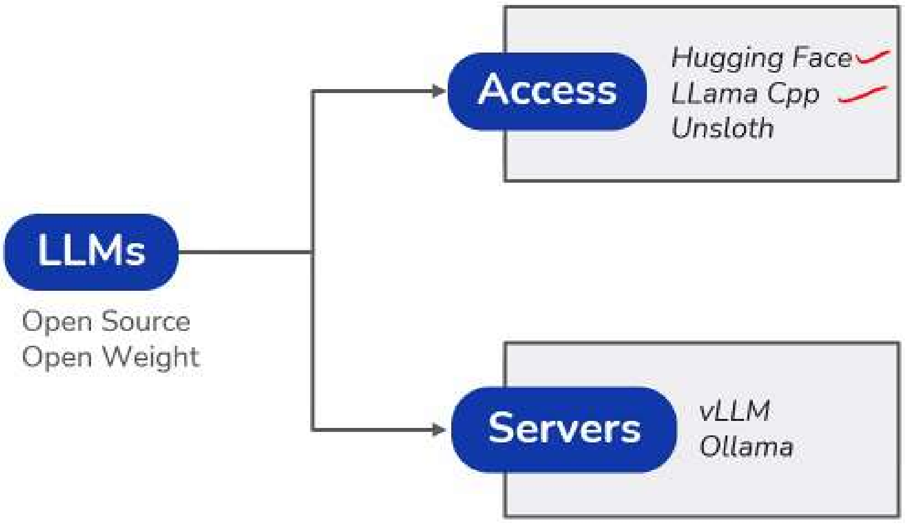
Accessing Large Language Models (LLMs)

LLMs (both paid and open source) can be accessed either through public cloud providers or LLM vendors.



Accessing Large Language Models (LLMs)

Open source/open weight LLMs can also be accessed using self-hosted company servers.



Efficient implementations of open models for specific target devices.

Efficient inference abstractions compatible with Open AI APIs

Comparing LLM Providers - Features

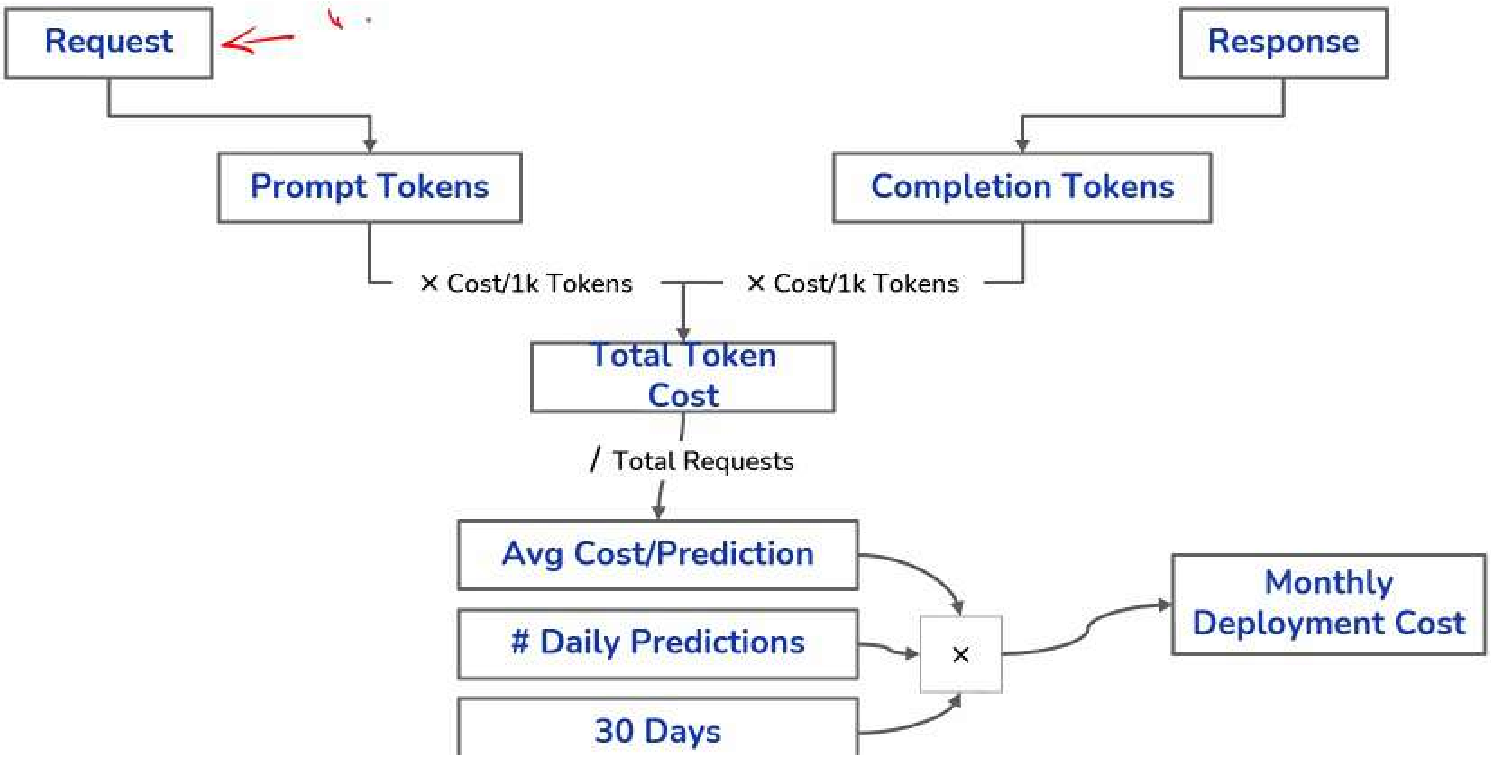
Feature	[✓ Azure]	AWS	[✓ GCP]
Exclusive model access	GPT (Open AI)	Claude (Anthropic)	Gemini Pro (Google)
Application ecosystem integration (e.g., LangChain, LlamaIndex)	High	Medium	Low
Stability & reliability of APIs	High	Medium	High
Customizability for production systems	High	Medium	High

Comparing LLM Providers - Cost Structures

Cost comparison per 1000 tokens

Model Type	Azure		AWS		GCP	
	Input	Output	Input	Output	Input	Output
Value-for-money, production grade (e.g., GPT3.5, Claude Instant, Gemini 1.5 Flash)	$0.0005	$0.0015	$0.0008	$0.0024	$0.000125	$0.000375
Advanced Reasoning Models (e.g., GPT4, Claude, Gemini 1.5 Pro)	$0.01	$0.03	$0.015	$0.075	$0.00125	$0.00375
Fine-tuned LLM variants	$0.0005	$0.0015				

Costing for LLM Projects



Request

Response

Prompt Tokens

Completion Tokens

× Cost/1k Tokens

× Cost/1k Tokens

Total Token Cost

/ Total Requests

Avg Cost/Prediction

Daily Predictions

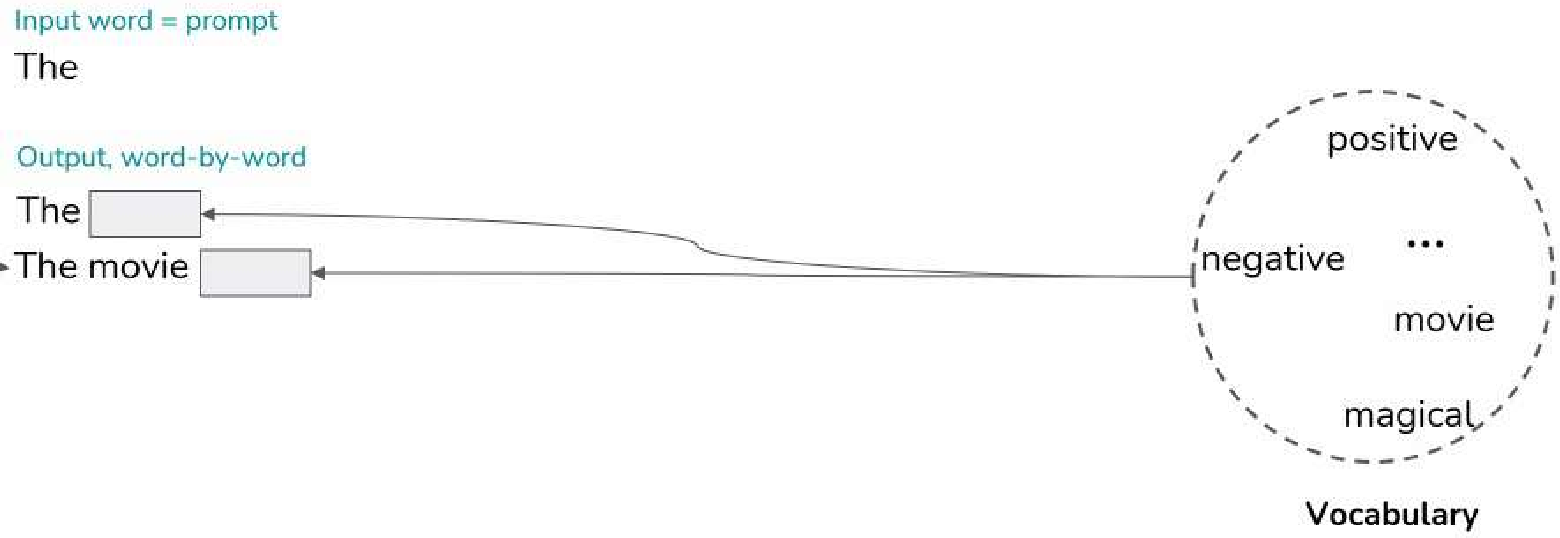
30 Days

×

Monthly Deployment Cost

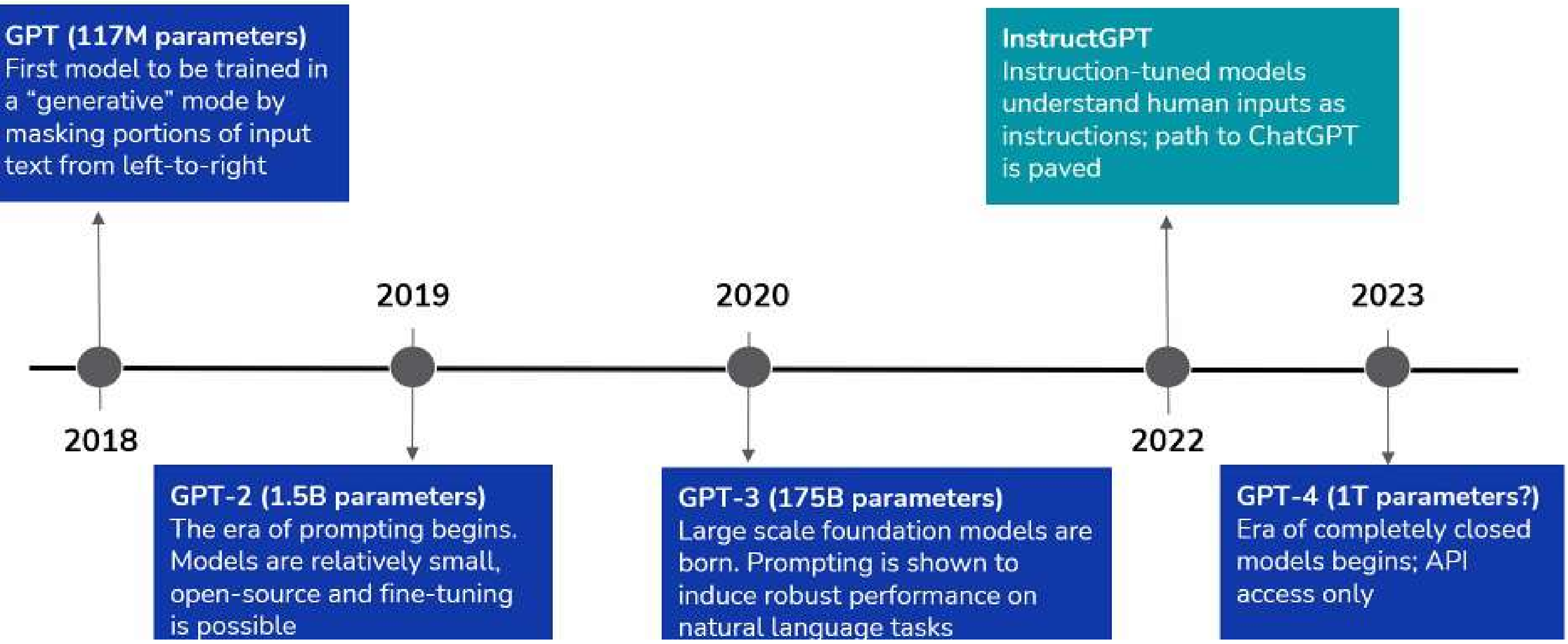
What are Large Language Models (LLMs)?

During inference, the LLM predicts the next word in the input sequence.



What are Large Language Models (LLMs)?

Over the last 2 years, LLMs (e.g., Open AI GPT) have evolved to be AI assistants



GPT (117M parameters)
First model to be trained in a "generative" mode by masking portions of input text from left-to-right

2018

2019

GPT-2 (1.5B parameters)
The era of prompting begins. Models are relatively small, open-source and fine-tuning is possible

2020

GPT-3 (175B parameters)
Large scale foundation models are born. Prompting is shown to induce robust performance on natural language tasks

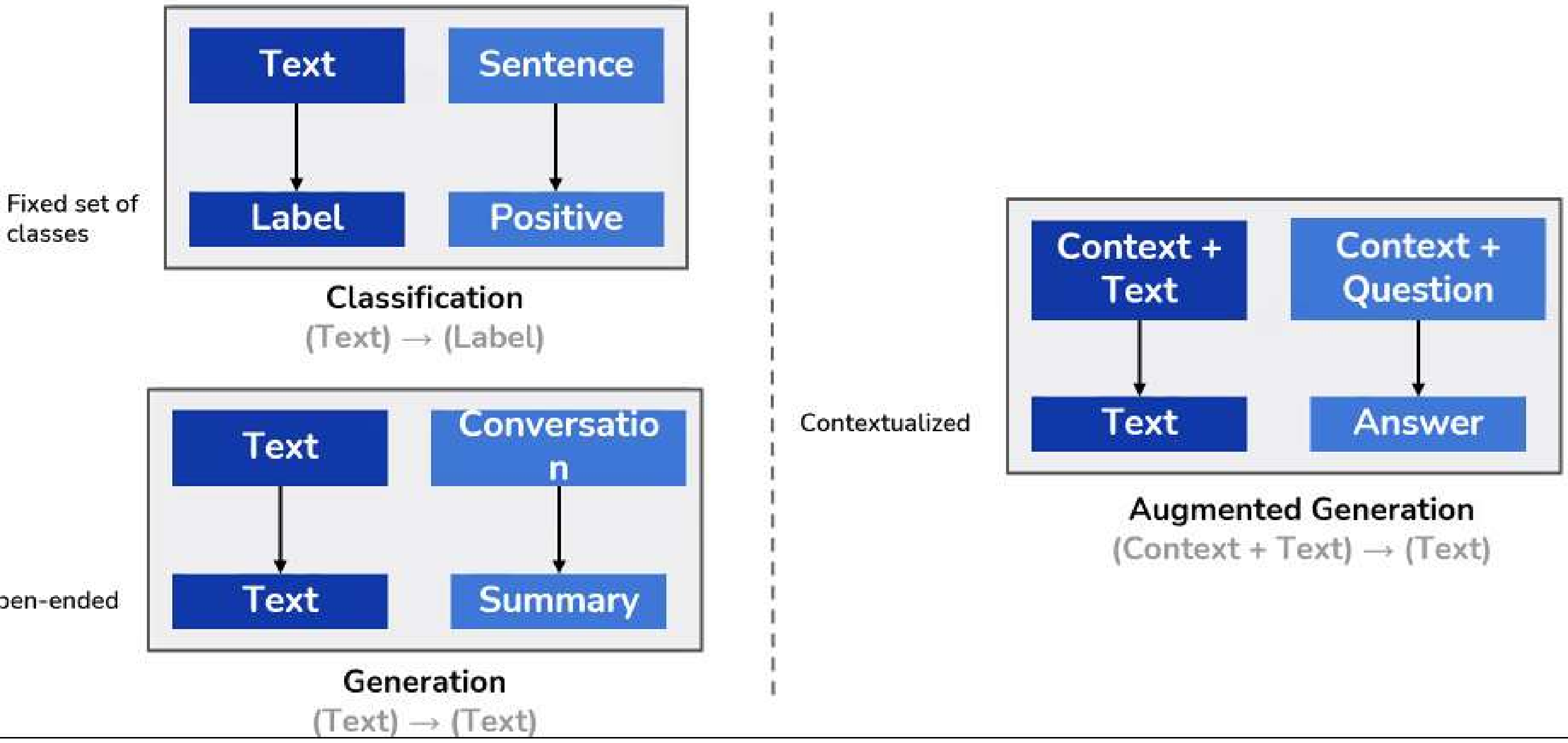
2022

InstructGPT
Instruction-tuned models understand human inputs as instructions; path to ChatGPT is paved

2023

GPT-4 (1T parameters?)
Era of completely closed models begins; API access only

Business Problems Solved by LLMs - A Taxonomy



Fixed set of classes

Text → Label

Sentence → Positive

Classification

(Text) → (Label)

pen-ended

Text → Text

Conversatio n → Summary

Generation

(Text) → (Text)

Contextualized

Context + Text → Text

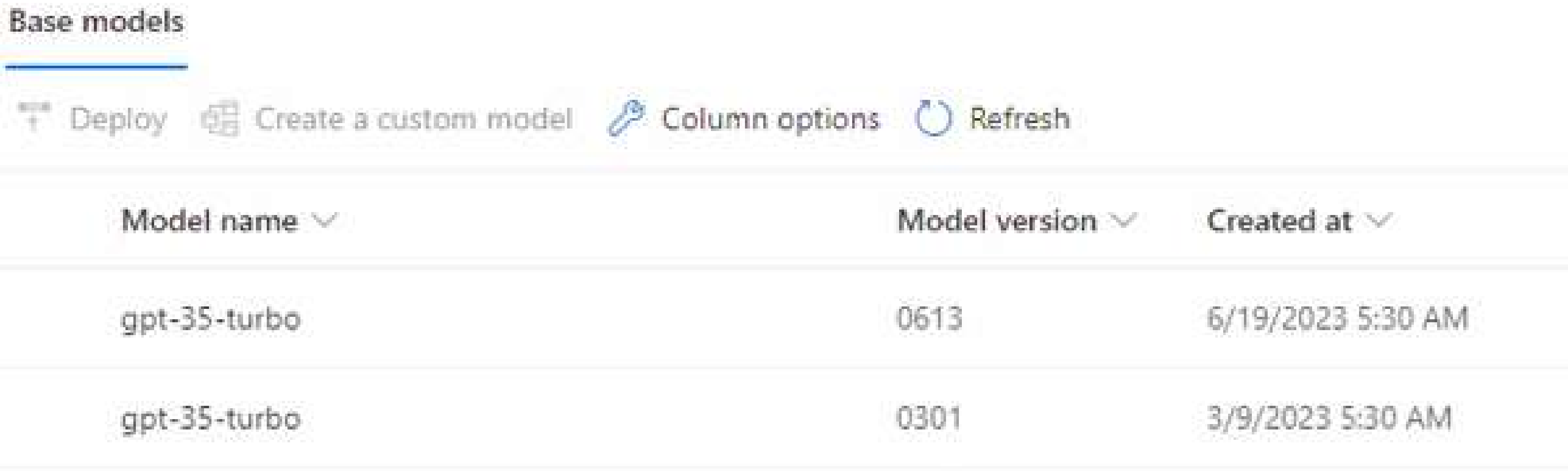
Context + Question → Answer

Augmented Generation

(Context + Text) → (Text)

Accessing LLMs using Azure Open AI

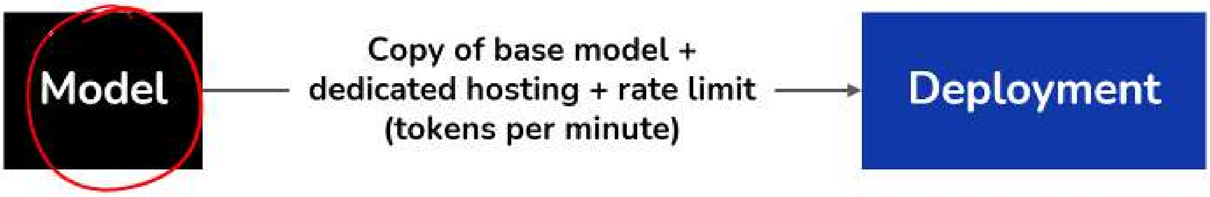
Azure provides controlled, fast access to a host of Open AI LLMs.

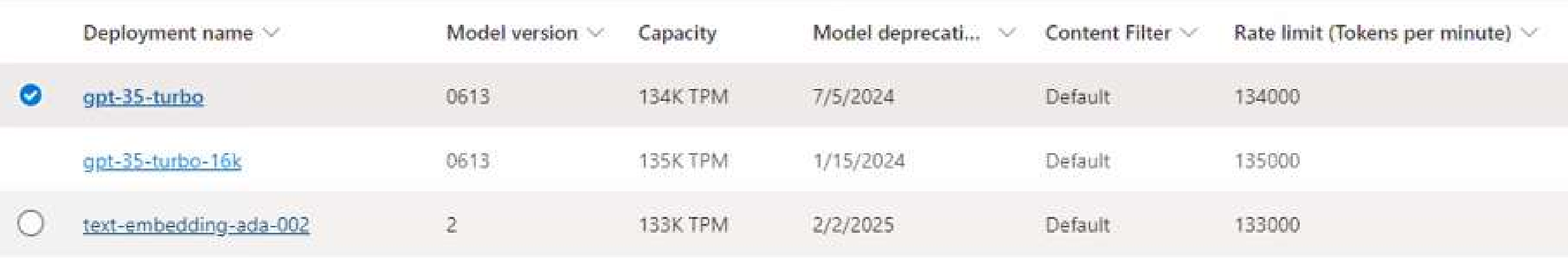


Model name	Model version	Created at
gpt-35-turbo	0613	6/19/2023 5:30 AM
gpt-35-turbo	0301	3/9/2023 5:30 AM

Accessing LLMs using Azure Open AI

Azure provides controlled, fast access to a host of Open AI LLMs.

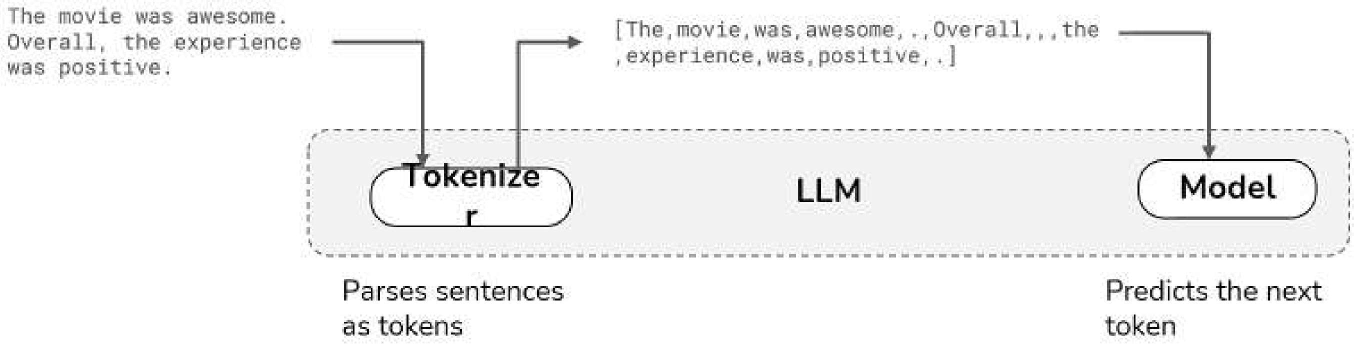

Model
Copy of base model + dedicated hosting + rate limit (tokens per minute)
Deployment



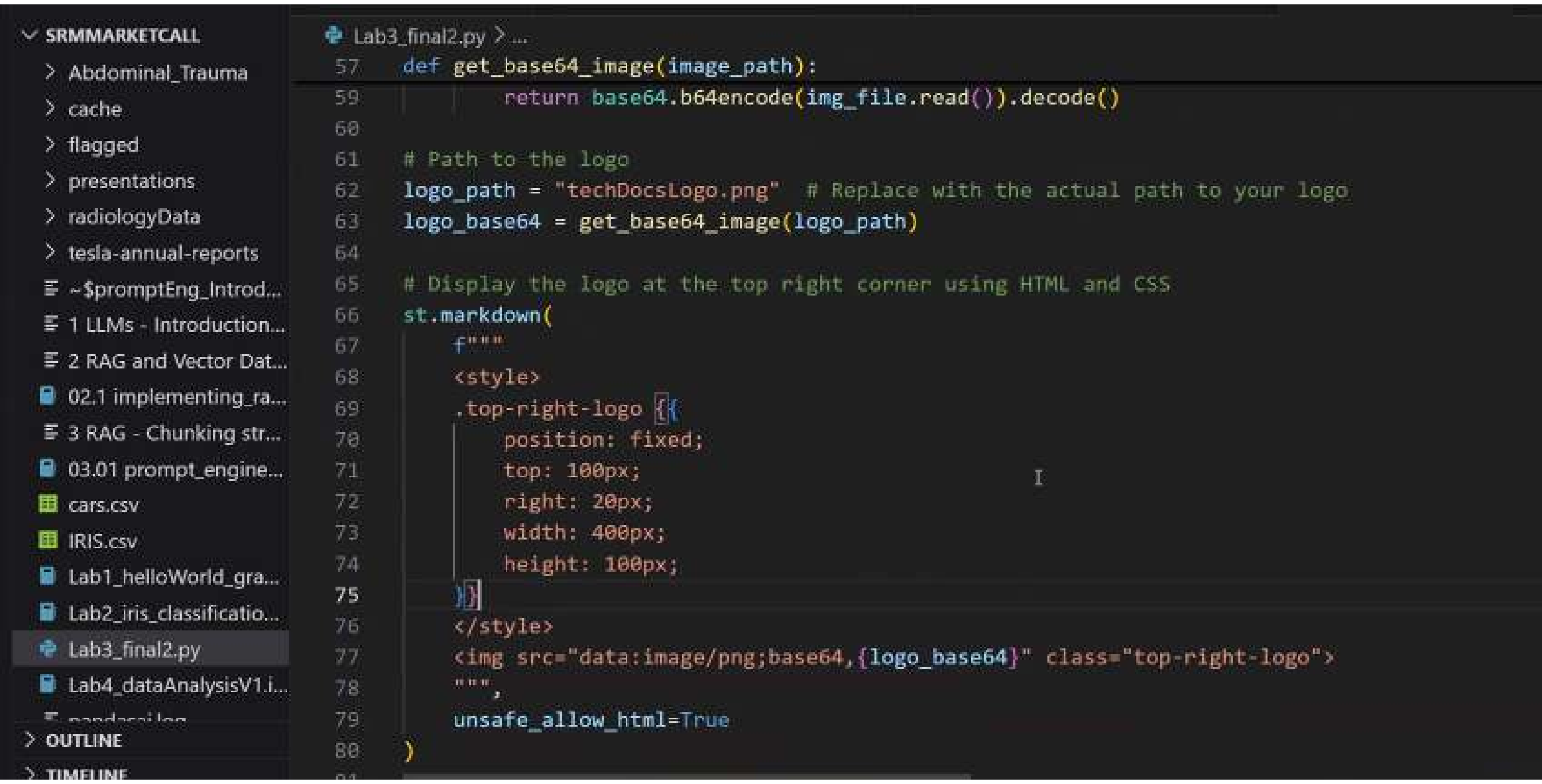
	Deployment name	Model version	Capacity	Model deprecati...	Content Filter	Rate limit (Tokens per minute)
✓	gpt-35-turbo	0613	134K TPM	7/5/2024	Default	134000
	gpt-35-turbo-16k	0613	135K TPM	1/15/2024	Default	135000
○	text-embedding-ada-002	2	133K TPM	2/2/2025	Default	133000

Understanding Tokens

A token refers to a segment or piece of text, such as a word, punctuation, or other meaningful element, into which input text is divided for processing by the model.



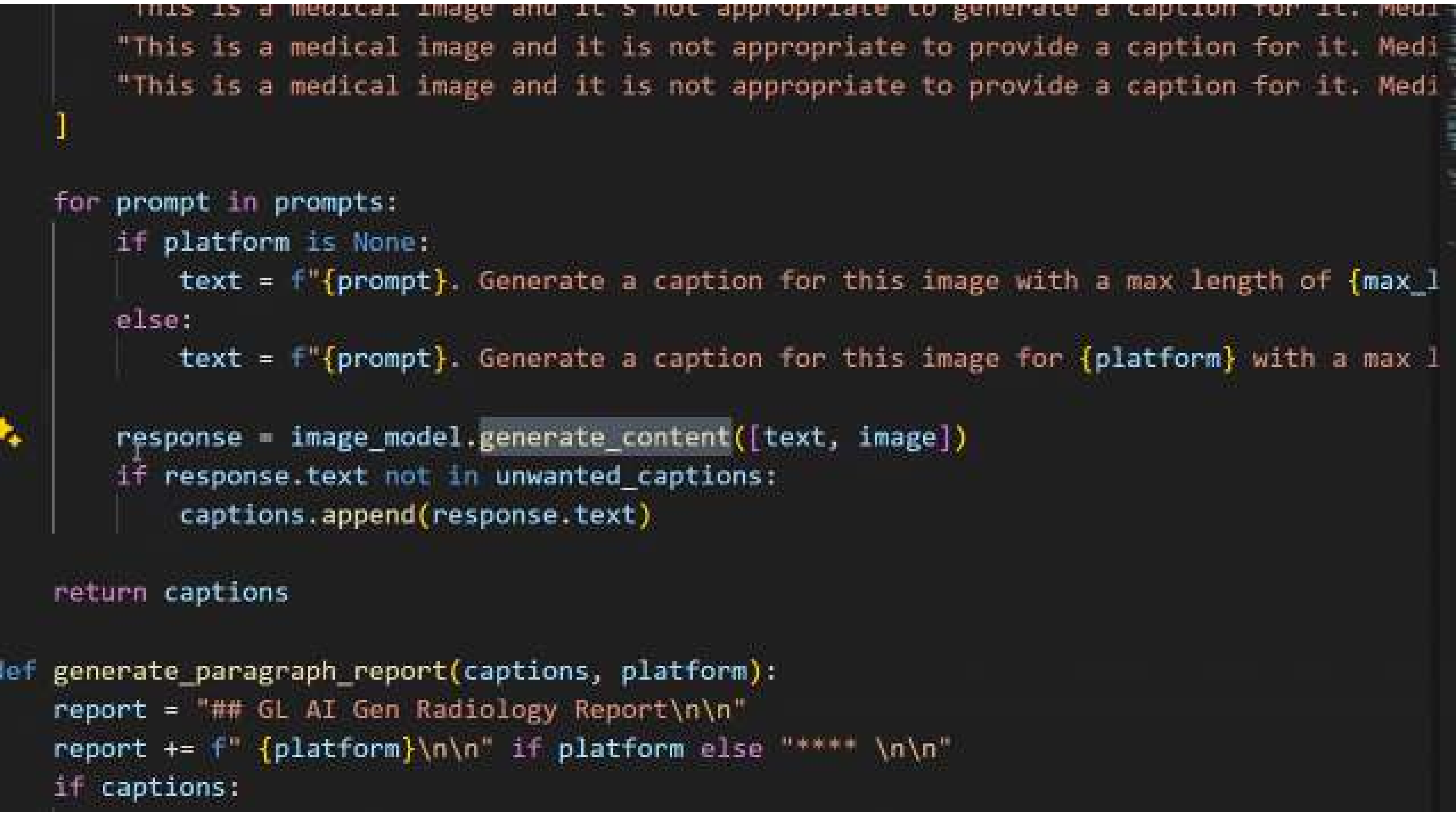
The movie was awesome. Overall, the experience was positive.
[The,movie,was,awesome,.,Overall,,,the ,experience,was,positive,.]
Tokenize r
LLM
Model
Parses sentences as tokens
Predicts the next token



```python
def get_base64_image(image_path):
        return base64.b64encode(img_file.read()).decode()

# Path to the logo
logo_path = "techDocsLogo.png"  # Replace with the actual path to your logo
logo_base64 = get_base64_image(logo_path)

# Display the logo at the top right corner using HTML and CSS
st.markdown(
    f"""
    <style>
    .top-right-logo {{
        position: fixed;
        top: 100px;
        right: 20px;
        width: 400px;
        height: 100px;
    }}
    </style>
    <img src="data:image/png;base64,{logo_base64}" class="top-right-logo">
    """,
    unsafe_allow_html=True
)
```



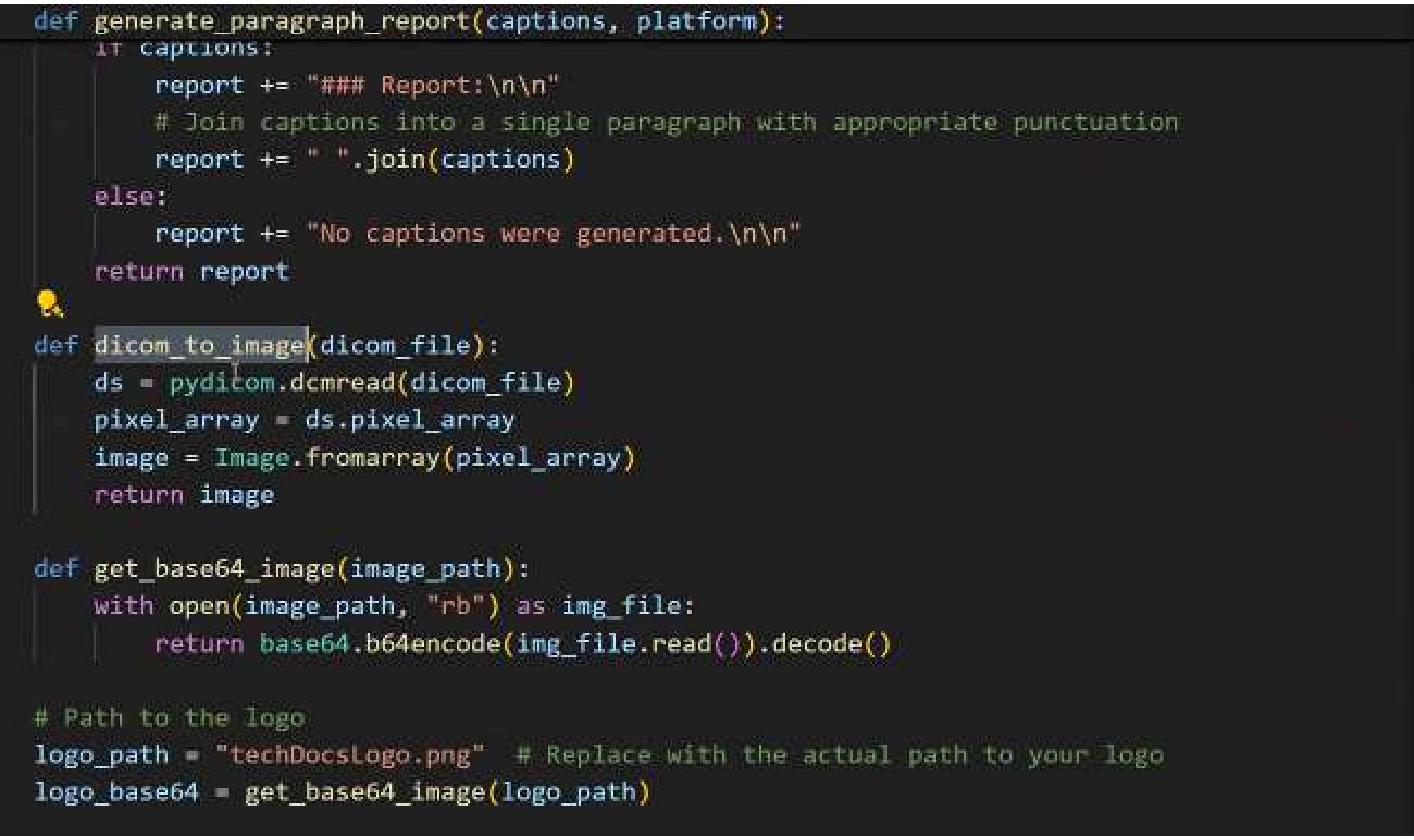
```
        "This is a medical image and it's not appropriate to generate a caption for it. Medi
        "This is a medical image and it is not appropriate to provide a caption for it. Medi
        "This is a medical image and it is not appropriate to provide a caption for it. Medi
    ]

    for prompt in prompts:
        if platform is None:
            text = f"{prompt}. Generate a caption for this image with a max length of {max_l
        else:
            text = f"{prompt}. Generate a caption for this image for {platform} with a max l

        response = image_model.generate_content([text, image])
        if response.text not in unwanted_captions:
            captions.append(response.text)

    return captions

ef generate_paragraph_report(captions, platform):
    report = "## GL AI Gen Radiology Report\n\n"
    report += f" {platform}\n\n" if platform else "**** \n\n"
    if captions:
```

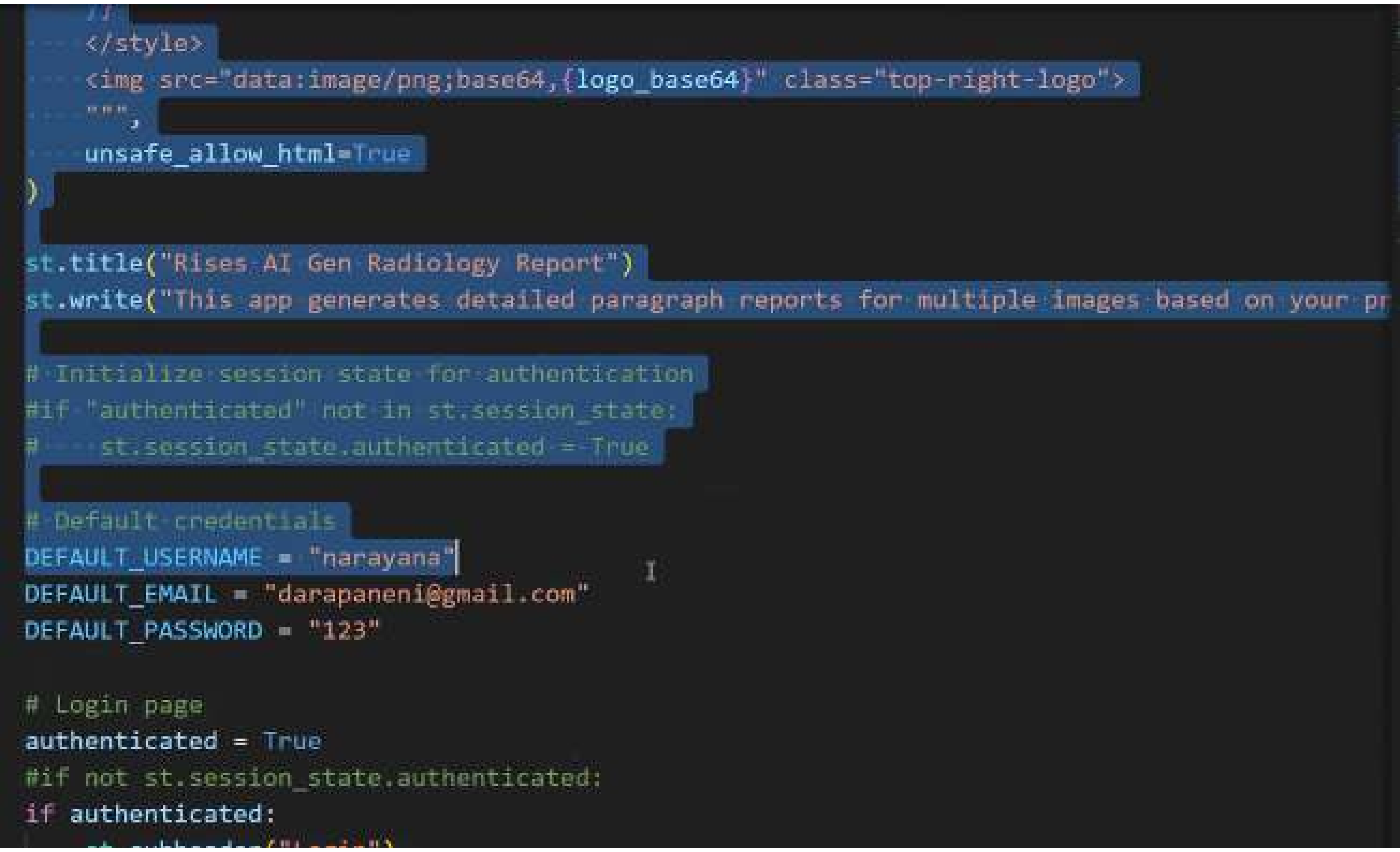


```python
def generate_paragraph_report(captions, platform):
    if captions:
        report += "### Report:\n\n"
        # Join captions into a single paragraph with appropriate punctuation
        report += " ".join(captions)
    else:
        report += "No captions were generated.\n\n"
    return report

def dicom_to_image(dicom_file):
    ds = pydicom.dcmread(dicom_file)
    pixel_array = ds.pixel_array
    image = Image.fromarray(pixel_array)
    return image

def get_base64_image(image_path):
    with open(image_path, "rb") as img_file:
        return base64.b64encode(img_file.read()).decode()

# Path to the logo
logo_path = "techDocsLogo.png"  # Replace with the actual path to your logo
logo_base64 = get_base64_image(logo_path)
```



```
    }}
    </style>
    <img src="data:image/png;base64,{logo_base64}" class="top-right-logo">
    """,
    unsafe_allow_html=True
)

st.title("Rises AI Gen Radiology Report")
st.write("This app generates detailed paragraph reports for multiple images based on your pr

# Initialize session state for authentication
#if "authenticated" not in st.session_state:
#    st.session_state.authenticated = True

# Default credentials
DEFAULT_USERNAME = "narayana"
DEFAULT_EMAIL = "darapaneni@gmail.com"
DEFAULT_PASSWORD = "123"

# Login page
authenticated = True
#if not st.session_state.authenticated:
if authenticated:
    st.subheader("Login")
```