Analyzing and Deploying Large Language Models in Corporate Environments

EIN4891C/ISC4323C Senior Design Capstone Project - Final Presentation

Presented To: All Points Logistics

Presented By: Team #9

Jacob Bandurski (Industrial Engineering) – Team Lead

Shraavani Bekkary (Data Science)

Maria Contractor (Data Science)

Jacques Coury (Industrial Engineering)

Kevin Dougherty (Data Science)

Faculty Mentors:

Dr. Mansooreh Mollaghasemi,

Dr. Luis Rabelo,

Dr. Ahmed Mohamed (Black Belt LSS)

Date: April 25th, 2025

Introduction and Project Purpose

- **All Points Logistics (APL):**
  - Service-Disabled Veteran-Owned Small Business (SDVOSB)
  - CMMI-DEV Level 3 Appraised; CMMC 2.0 Level 2 Assessed
  - Core Capabilities: IT Services & Consulting, Software Development, Engineering, Cyber Security, Integrated Logistics
  - Clientele: Primarily US Government, Department of Defense (DoD), and NASA.
    - Often mission-critical projects demanding high security and compliance.
- **The LLM Opportunity & Challenge:**
  - Rapid advancements in large language models (LLMs) offer the potential to enhance APL’s capabilities and efficiency.
  - Deployment within APL's high-compliance environment requires careful, structured evaluation to manage significant security, compliance, and operational risks.

- **Project Purpose:** To systematically analyze the LLM landscape to deliver data-driven, justified recommendations and an actionable deployment roadmap tailored to APL's needs and constraints.
- **Team:** *Interdisciplinary* collaboration between UCF Industrial Engineering (IE) and Data Science (DS) students.

Source: https://allpointsllc.com/capabilities/

Problem Statement

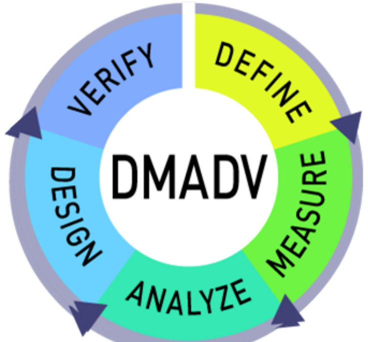
- **Context:** APL operates in a demanding high-compliance (CMMI L3, CMMC L2) and high-security environment serving critical Gov/Defense/Aerospace clients.
- **Challenge:** While LLMs offer potential benefits, APL seeks a structured, data-driven approach to:
  - Systematically evaluate the diverse LLM landscape.
  - Assess model capabilities against compliance requirements (e.g., CUI handling).
  - Quantify the value proposition reliably.
  - Define a secure, compliant, and effective deployment strategy.
- **Risk:** Without this approach, APL risks missing strategic opportunities or making suboptimal adoption decisions that could compromise security, compliance (CMMC/CMMI), mission support, or operational performance.
- **Overall:** The team aims to enhance All Points Logistics' business operations by comprehensively analyzing 12 large language models (LLMs) and evaluating their characteristics through research, prompt engineering, and metrics.

Source: https://allpointsllc.com/contract-vehicles/

Project Goal & Objectives

- **Goal:** By April 28th, this project will deliver:
  - A comprehensive LLM analysis.
  - Actionable deployment roadmap, and justified recommendations for 1-3 suitable LLMs for APL, demonstrating evaluated potential for business use cases.
  - Data-driven recommendations will align security/compliance needs and be supported by deliverables accepted by APL/UCF faculty according to the project timeline.
- **Main Objective:** Identify high-value LLM use cases within APL's capabilities.
  - Evaluate LLM potential for Software Development productivity gains (code gen/test/debug).
  - Assess LLM capability for Launch Support scheduling decision support/analysis.
  - Determine LLM suitability for compliant, parameter-based Solution Identification (e.g., parts, products).
- **Scope Summary:**
  - **In Scope:** LLM Taxonomy; In-depth Analysis (~12 LLMs vs. CTQs); Down-selection; Targeted Experimentation (Prompting); Deployment Roadmap Design; Final Recommendations.
  - **Out of Scope:** Full Production Implementation/Integration; LLM License/Infrastructure Procurement for APL; Production Fine-tuning; Custom Application Development.

Methodology: DMADV Framework



- **Framework Choice:** DMADV (Define, Measure, Analyze, Design, Verify) is selected as appropriate for *designing* new capabilities/processes, aligning with Design for Six Sigma (DFSS).

- **Phase Goals for This Project:**
  - **Define:** Establish project foundation, goals, scope, and critical APL requirements (CTQs).
  - **Measure:** Systematically collect data characterizing | 12 LLM candidates against CTQS ("AS-IS" of options).
  - **Analyze:** Evaluate LLMs vs. CTQs, identify tradeoffs, select candidates.
  - **Design:** Develop a ”TO-BE” deployment strategy/roadmap for selected LLM(s).
  - **Verify:** Validate the final design against CTQS and plan for control.

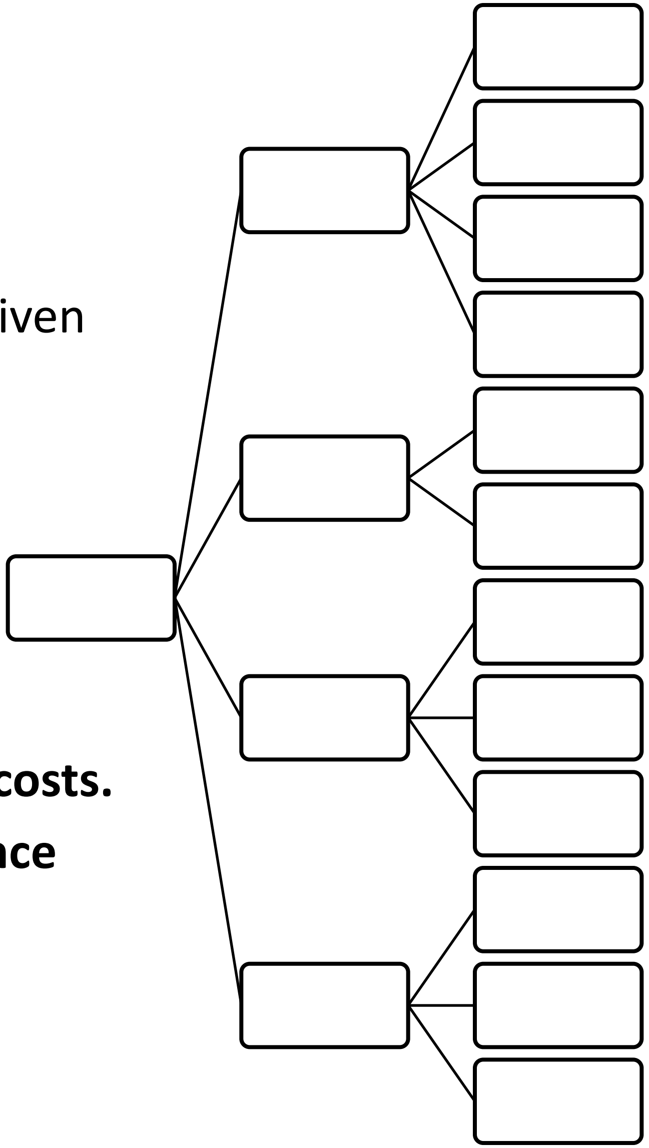
- **Key Tools Applied (Conceptual/Actual):** SIPOC, Process Maps (PMAPs), Voice of Customer (VOC) Analysis, Critical-to-Quality (CTQ) Trees, Multi-Criteria Decision Analysis (MCDA), Gap Analysis, Cost-Benefit Analysis (CBA/TCO), Failure Modes & Effects Analysis (FMEA), Risk Register, Stakeholder Analysis, Control Plan, Requirements Traceability Matrix.

Source: https://asq.org/training/design-for-six-sigma--dfss--dfssasq?srsltid=AfmBOore8EXaE0HWP9IedjZ8mGq3e-ztpVKGOsc1117mDUUtwQ_Ca92x

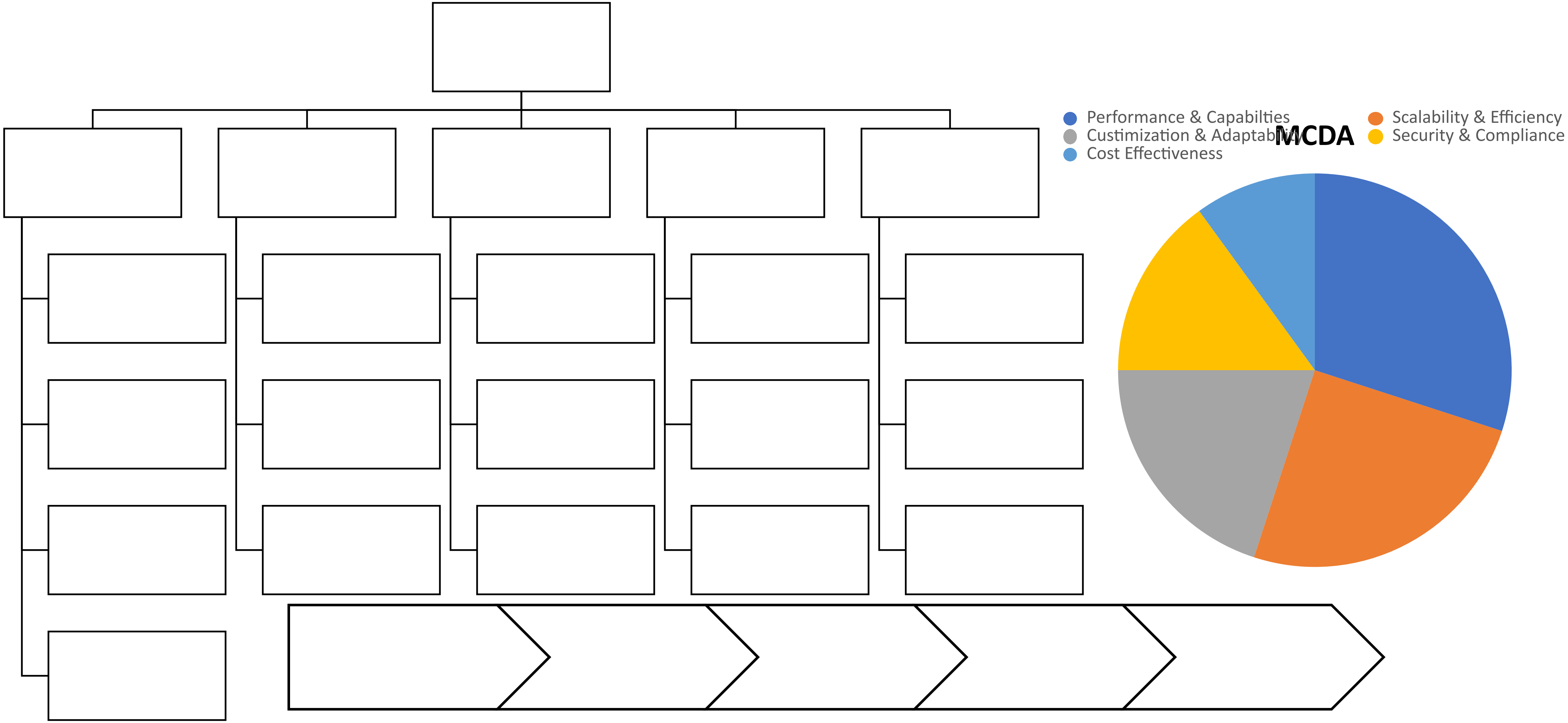
CTQ: Value Proposition

Challenge: While LLMs offer potential benefits, APL seeks a structured, data-driven approach to:

- Systematically evaluate the diverse LLM landscape.
- Vendor Pricing Model -> **Analyze model clarity & predictability; Include potential GovCloud premiums.**
- Infrastructure Costs -> **Model costs accurately for required compliant infrastructure.**
- Implementation Effort -> **Estimate and include labor & specialized skill costs.**
- Ongoing Operational Costs -> **Project recurring costs including compliance sustainment.**



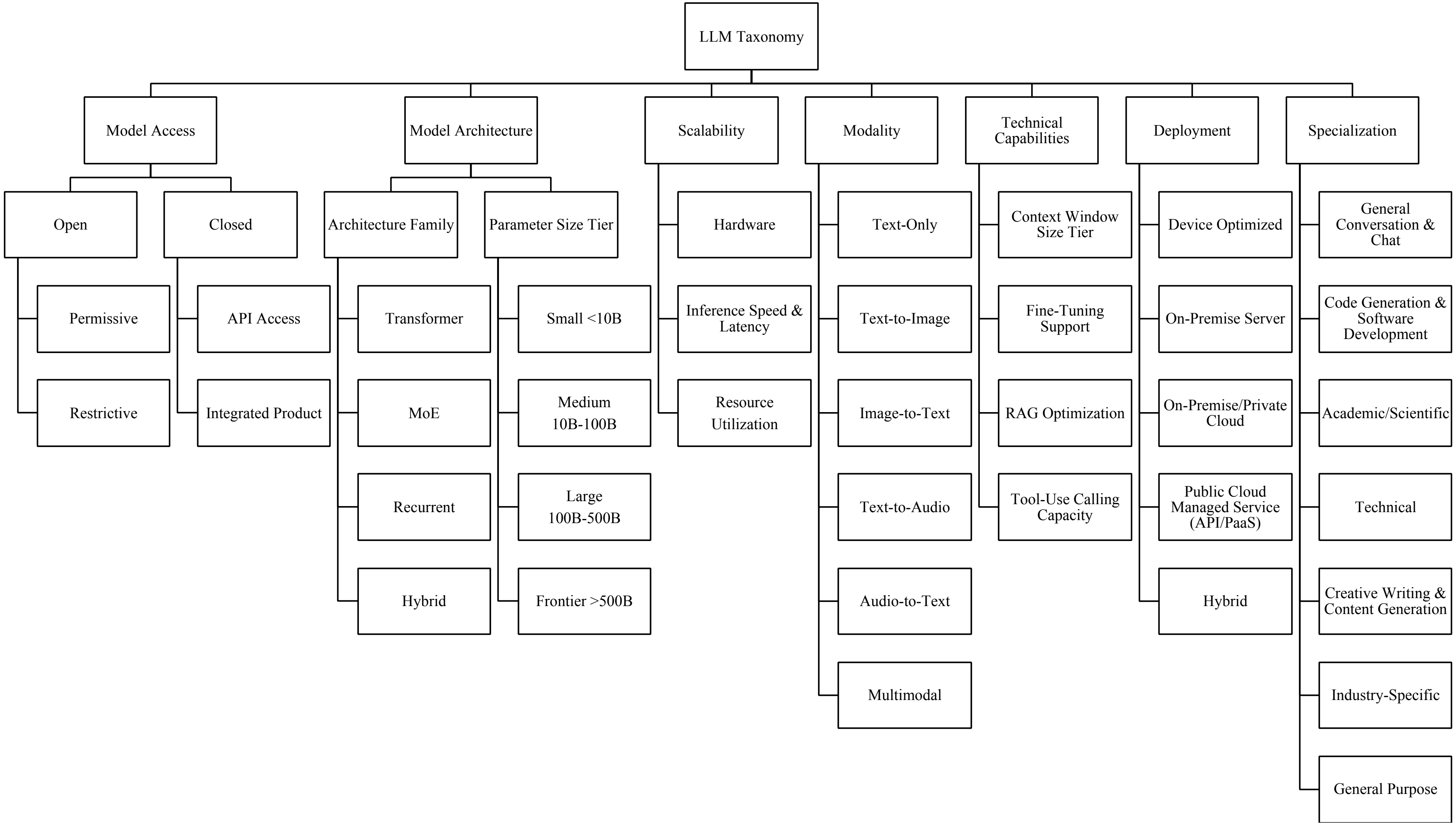
Operations Research


● Performance & Capabilties
● Custimization & Adaptability
● Cost Effectiveness
● Scalability & Efficiency
● Security & Compliance
MCDA

*Final Score=(0.15×Performance)+(0.2×Scalability)+(0.1×Customization)+(0.3×Security)+(0.25×Cost)

Findings/Analysis

LLM taxonomy



- LLM Taxonomy
  - Model Access
    - Open
      - Permissive
      - Restrictive
    - Closed
      - API Access
      - Integrated Product
  - Model Architecture
    - Architecture Family
      - Transformer
      - MoE
      - Recurrent
      - Hybrid
    - Parameter Size Tier
      - Small <10B
      - Medium 10B-100B
      - Large 100B-500B
      - Frontier >500B
  - Scalability
    - Hardware
    - Inference Speed & Latency
    - Resource Utilization
  - Modality
    - Text-Only
    - Text-to-Image
    - Image-to-Text
    - Text-to-Audio
    - Audio-to-Text
    - Multimodal
  - Technical Capabilities
    - Context Window Size Tier
    - Fine-Tuning Support
    - RAG Optimization
    - Tool-Use Calling Capacity
  - Deployment
    - Device Optimized
    - On-Premise Server
    - On-Premise/Private Cloud
    - Public Cloud Managed Service (API/PaaS)
    - Hybrid
  - Specialization
    - General Conversation & Chat
    - Code Generation & Software Development
    - Academic/Scientific
    - Technical
    - Creative Writing & Content Generation
    - Industry-Specific
    - General Purpose

GPT-4

Baseline

- Model: GPT-4 by OpenAI
- Use Case Focus: General-purpose enterprise LLM — excels in code generation, customer support, research summaries, and language-heavy workflows.
- Tested On: OpenAI API (GPT-4 turbo)
- Ideal Fit For: Teams needing high-quality, reliable outputs with strong reasoning ability.
- Vendor Lock-in: Yes — API-only, hosted by OpenAI. No open weights or local deployment.
- Key Takeaway: GPT-4 is extremely capable and consistent across almost all use cases but is expensive and cannot be fine-tuned or hosted locally.

Performance and Setup

- Memory Footprint: N/A (hosted by OpenAI)
- Runtime (Inference):
- API response time: ~1–3 seconds
- Fine-Tuning: Not currently supported
- Setup Time: ~10 mins to set up API key and environment (e.g., Google Colab, Postman, or Python requests library)

Cost

- **Inference token cost**:
- GPT-4 Turbo: ~$0.01 per 1K tokens (input), ~$0.03 per 1K tokens (output)
- **Access**: Requires OpenAI account and API key

Deployment Type	Description	Cost (Hard)	Setup Complexity
OpenAI API	Fully Hosted	Pay-per-Use	Low

Pros and Cons

- Supports:
  - Natural language reasoning
  - Multilingual understanding
  - Code generation (Python, SQL, JS, etc.)
  - Summarization and chatbot functionality
- Does Not Support Natively:
  - Local hosting or open-weight usage
  - Image or vision tasks (requires GPT-4V)
  - Custom fine-tuning
- Security Considerations:
  - Data goes through OpenAI’s hosted environment (make sure to use dummy data for sensitive content)

Summary

- **GPT-4** is best for teams that want high-quality outputs and are okay with vendor lock-in.
- Not ideal for companies needing local control or fine-tuning, but unbeatable for overall reasoning and reliability.
- **Next Steps**: Compare side-by-side with other LLMs to determine best one for different departments.

Claude (Anthropic)

Baseline

- **Model**: Claude 2.1 (by Anthropic)
- **Use Case Focus**: Internal communication, HR automation, document summarization, compliance tasks
- **Tested On**: Anthropic API (via web or SDK)
- **Ideal Fit For**: Teams prioritizing safety, transparency, and explainability in AI
- **Vendor Lock-in**: Yes — Claude is only accessible via Anthropic's API
- **Key Takeaway**: Claude is ideal for sensitive workflows where response safety and transparency matter. It is less prone to toxic output but slightly weaker in technical reasoning tasks.

Performance and Setup

- **Memory Footprint**: N/A (hosted service)
- **Runtime (Inference)**:
- ~1–2 seconds average response time via API
- **Fine-Tuning**: Not supported (can do prompt tuning only)
- **Setup Time**: ~5–10 minutes to configure Anthropic API with an API key

Cost

- **Inference token cost**:
- Input: ~$0.008 / 1K tokens
- Output: ~$0.024 / 1K tokens
- **Access**: Requires API key (request from Anthropic)

Deployment Type	Description	Cost (Hard)	Setup Complexity
Anthropic API	Fully Hosted, web based	~$0.008–0.024/1K tokens	Low

Pros and Cons

- Supports:
  - Strong on safe, transparent outputs
  - Handles structured prompts well
  - High context window (up to 200K tokens)
  - Good for HR, legal, and policy generation
- Does Not Support Natively:
  - Local hosting or open weights
  - Image or multimodal inputs
  - Deep technical or math-heavy prompts
- Security Considerations:
  - Anthropic enforces strong safety guardrails
  - Ideal for compliance-focused environments

Summary

- Claude is best for use cases that demand ethical AI and low-risk outputs—such as HR, legal, or sensitive corporate tasks. While it’s not as strong as GPT-4 for complex reasoning, its long context window and safe generation make it a strong enterprise option.
- **Next Steps**: Compare side-by-side with other LLMs to determine best one for different departments.

PaLM 2 (Google)

Baseline

- **Model**: PaLM 2 (by Google, used in Bard)
- **Use Case Focus**: Multilingual communication, lightweight coding tasks, real-time productivity tools (e.g., document editing)
- **Tested On**: Google Vertex AI and Bard
- **Ideal Fit For**: Teams needing real-time assistance, lightweight multilingual tasks, and seamless integration with Google Cloud
- **Vendor Lock-in**: Yes — hosted by Google Cloud, not open-source
- **Key Takeaway**: PaLM 2 is a fast and scalable LLM with strengths in translation, language generation, and code. Ideal for companies already in the Google ecosystem.

Performance and Setup

- **Memory Footprint**: N/A (Google-hosted)
- **Runtime (Inference)**:
- Fast response (~1 sec) in Bard or Vertex AI
- **Fine-Tuning**: Not directly supported for PaLM 2 (limited prompt tuning only)
- **Setup Time**: 5–10 mins if using Google Cloud or Vertex AI

Cost

- **Inference cost (Vertex AI)**:
- ~$0.005–0.01 / 1K tokens (varies by instance type and plan)
- **Access**: Google Cloud account required

Deployment Type	Description	Cost (Hard)	Setup Complexity
Google Vertex AI	Hosted in Google Cloud	~$0.005–$0.01 / 1K tokens	Low

Pros and Cons

- Supports:
    - Multilingual outputs
    - Fast inference speed
    - Tight integration with Google Docs, Sheets, etc.
    - Strong for translation, basic reasoning, and coding
- Does Not Support Natively:
    - Open-weight/local deployment
    - Complex reasoning or very long context like Claude or GPT-4
    - Real-time fine-tuning
- Security Considerations:
    - Built-in privacy and security aligned with Google Cloud standards
    - Good for orgs already using Google products

Summary

- PaLM 2 is great for enterprises using Google Cloud and requiring quick, multilingual, and low-cost outputs. Its productivity integrations and fast performance make it suitable for customer service, translation, or content generation.
- **Next Steps**: Compare side-by-side with other LLMs to determine best one for different departments.

Copilot (Microsoft)

Microsoft 365
Copilot

Baseline

- **Model:** Built upon OpenAI's GPT-4 foundational models (incl. Turbo/GPT-4o variants); Integrated via Microsoft's proprietary "Prometheus" framework for M365 tasks.
  - **Use Case Focus:** Enterprise productivity within Microsoft 365: Document drafting/summarization/rewriting (Word), data analysis/ visualization (Excel), presentation creation (PowerPoint), email assistance (Outlook), meeting summarization/Q&A (Teams).
  - **Tested On:** Underlying models benchmarked externally; Integrated performance validated via Microsoft's internal testing, Responsible AI protocols, user previews, and ongoing feedback within the M365 application/Graph context.
  - **Ideal Fit For:** Organizations heavily invested in the Microsoft 365 ecosystem (SharePoint, OneDrive, Teams, Office Apps) needing deeply integrated AI assistance grounded in organizational data, operating within Microsoft's security/compliance framework (incl. GCC High for Gov/Defense like APL).
  - **Key Takeaway:** A powerful, integrated AI "co-pilot" boosting productivity by securely leveraging organizational data within the M365 environment; not a standalone general-purpose LLM API.
- **Performance and Setup:**
  - **Memory Footprint:** Not Applicable / Disclosed (Managed by Microsoft SaaS).
  - **Runtime (Inference):** Highly Variable / Not Specified (Managed Service). It depends on prompt, Graph retrieval load, and M365 app integration context.
  - **Runtime (Fine-tuning):** Not Applicable (Customization via RAG/Graph grounding & Copilot Studio extensions).
  - **Soft Cost:** Significant (Requires extensive Data Governance prep [permissions, labels], robust Change Management [training, policy], M365 Administration, IT Readiness).
  - **Inference Token Cost:** Bundled in Subscription (~$30/user/month add-on to qualifying M365 license). Not metered per token.
  - **Access:** Requires qualifying M365 Commercial/Education license + paid Copilot add-on license per user.

Source: Microsoft Learn. (n.d.). *Data, Privacy, and Security for Microsoft 365 Copilot*. Microsoft Learn. Retrieved April 25, 2025, from https://learn.microsoft.com/en-us/copilot/microsoft
microsoft-365-copilot-privacy

Baseline

- **Pros/Cons and Security:**

  Supports:

  - Deep M365 app integration & Graph data grounding.
  - Leverages powerful GPT-4 class models.
  - Enterprise-grade data protection (no training on org data).
  - Operates within M365 security/compliance boundaries (inherits policies).
  - **FedRAMP High path** (via M365 GCC High offering).
  - Centralized M365 administration & reporting.
  - Extensibility via Microsoft Copilot Studio.

  Does Not Support Natively:

  - Direct base model fine-tuning by customer.
  - Standalone API access for general development outside Copilot Studio framework.
  - Guaranteed/fixed performance SLAs (managed service).
  - Potentially limited direct input modalities vs. consumer versions.
  - Accessing data user lacks M365 permissions for.

  Security Consideration:

  - **Permissions Paramount:** Effectiveness and safety rely heavily on correctly configured M365 permissions to prevent unintended internal data exposure.
  - **Data Governance Prerequisite:** Requires mature SharePoint/OneDrive permissions, sensitivity labeling, and DLP policies.
  - **Standard LLM Risks:** Prompt injection, reliance on output accuracy (mitigated by grounding but still requires user validation).
  - **Plugin Risk:** Security of third-party connections via Copilot Studio must be vetted.
  - **Vendor Dependency:** Relies entirely on Microsoft's platform and security practices.

- **Summary:** This solution offers unparalleled M365 integration and contextual assistance for securely using organizational data.
  - Its strengths are deep integration and a robust enterprise security/compliance framework (including GovCloud).
  - It requires significant data governance readiness and is best suited for organizations embedded in the M365 ecosystem.
  - The cost is subscription-based per user.

Source: Microsoft. *AI for Enterprise Productivity | Microsoft 365 Copilot*. Microsoft. Retrieved April 5, 2025, from https://www.microsoft.com/en-us/microsoft-365/copilot/enterprise

Gemini (Google)

Baseline

- **Model:** Google Gemini family (Pro tier); transformer-based, natively multimodal. (The Parameter count has not been disclosed.)
    - **Use Case Focus:** High-performance general AI tasks: complex reasoning, coding assistance, long-document analysis (via large context window), multimodal understanding (text, image, audio, video), and enterprise tasks via platform integration.
    - **Tested On:** Extensive internal evaluations; Strong public benchmark performance reported (MMLU, HumanEval, MATH, etc.); Real-world use in Google products.
    - **Ideal Fit For:** Organizations needing state-of-the-art multimodal & long-context capabilities (1M/2M tokens with 1.5 Pro), operating within GCP, or requiring high compliance (e.g., FedRAMP High via Vertex AI/Workspace).
    - **Key Takeaway:** Frontier model family with leading multimodal/long-context features, accessible via Google's secure/ compliant cloud infrastructure.

- **Performance and Setup:**
    - **Memory Footprint:** N/A (Managed by Google for API/PaaS access).
    - **Runtime (Inference):** Variable (Platform Managed); depends on the model, query complexity, and load. Benchmarks suggest competitive latency/throughput.
    - **Runtime (Fine-tuning):** Variable (Via Vertex AI); depends on data, configuration, and compute resources within GCP.
    - **Soft Cost:** Significant (Requires GCP expertise, data prep for RAG/tuning, integration effort, prompt engineering, training, and change management).
    - **Inference Token Cost:** Per Token (via Google AI/Vertex AI APIs—rates vary, e.g., ~$1.25/$5.00 per 1M I/O for 1.5 Pro); per user subscription (via Google Workspace integration).
    - **Access:** Google AI Studio API (free tier possible), Google Cloud Vertex AI (requires GCP billing), and Google Workspace (requires qualifying license + Gemini add-on).

Source: Google AI for Developers. *Gemini Developer API Pricing* | *Gemini API*. Google AI for Developers. Retrieved April 7, 2025, from https://ai.google.dev/gemini-api/docs/pricing

Baseline

- **Pros/Cons and Security:**

  Supports:

  - State-of-the-art benchmark performance (reasoning, coding).
  - Native Multimodality (text, image, audio, video).
  - Exceptional Context Windows (up to 1M/2M tokens - 1.5 Pro).
  - Robust GCP Infrastructure (scalability, MLOps via Vertex AI).
  - **FedRAMP High path** (via Vertex AI Assured Workloads / Workspace).
  - Fine-tuning capabilities (within Vertex AI).
  - Optional grounding via Google Search.
  - Strong inherited GCP security/compliance posture (SOC 2, ISO, HIPAA BAA support).

  Does Not Support Natively:

  - Open-source self-hosting (for Pro models).
  - Deep out-of-the-box integration with non-Google ecosystems (e.g., M365)
  - Guaranteed, fixed latency (performance is managed by service)

  Security Consideration:

  - It relies on robust GCP security and proper customer configuration (IAM, VPC-SC, Data Governance).
  - GCP terms govern data privacy; enterprise controls typically prevent data use for model training.
  - Secure data management is essential for RAG/fine-tuning within GCP.
  - Standard LLM risks (hallucination, bias, prompt injection) require mitigation.
  - Vendor dependency on Google Cloud ecosystem.

- **Summary:**
  - Cutting-edge multimodal and long-context model via Google's secure/compliant cloud (Vertex AI/Workspace),
  - ideal for complex tasks within GCP or high-compliance needs.
  - It requires cloud expertise and careful governance and offers consumption or subscription pricing.

Source: Google Workspace Blog. (2025, March 18). *Gemini in Workspace achieves FedRAMP High authorization*. Google Workspace Blog. Retrieved April 7, 2025, from https://workspace.google.com/blog/identity-and-security/gemini-workspace-apps-and-gemini-app-are-first-achieve-fedramp-high-authorization

Mistral

Baseline

- Model: Mistral-7B-Instruct-v0.1
- Tested On: Google Colab (T4 GPU)
- Ideal Fit For: Companies seeking control, fine-tuning capability, and flexible deployment without vendor lock-in
- By no vendor lock-in we mean:
  - Open-weight (you can download and run them yourself)
  - Fine-tunable
  - Deployable anywhere:
    - Your own GPU server
    - Any cloud (AWS, GCP, Azure)
    - Even air-gapped environments (Isolated from any internet or unsecured networks)
- Key Takeaway: Mistral provides strong inference capabilities, competitive cost structure, and customizable deployment models (cloud/on-prem)

Performance and Setup

- Memory Footprint: ~4 GB+ for inference, 40+ GB recommended for fine-tuning
- Runtime (Inference):
  - Cold boot: ~1.5 minutes to load
  - Inference: <10 seconds on small input (3-sensor values + structured prompt)
- Runtime (Fine-Tuning): Requires multi-hour sessions & custom dataset + PEFT tools LoRA(Low-rank adaptation), QLoRA
- Soft Cost: ~3 hours to set up environment, install packages, and handle token gating, and ~2 minutes per runtime restart.

Cost

- Inference token cost: None (if using open weights locally)
- Access: Requires Hugging Face login and gated model request approval

Deployment Type	Description	Cost (Hard)	Setup Complexity
Colab Pro+ (GPU)	Pay-as-you-go, fast setup	$49/month	Low
AWS EC2 (A100)	Scalable, flexible	~$1.50/hr	Medium
On-Prem (Consumer GPU)	RTX 4090 w/ 24GB VRAM	~$1,800+	High
On-Prem (Server)	A100/H100 server	$10,000–$30,000	High

Pros/Cons

- Supports:
    - Inference on structured and natural language prompts
    - Fine-tuning with LoRA
    - Context size: ~32k tokens (~24k words or ~50 pages of text) ideal for structured CSV summaries
- Does Not Support Natively:
    - Image generation or image input (use paired vision encoders instead)
        - Paired models like CLIP (Contrastive Language–Image Pretraining), LLaVA (Large Language and Vision Assistant) can be used externally for multimodal tasks.
- Security Considerations:
    - Full local control possible
    - Can run air-gapped for sensitive data environments

Summary

- Mistral-7B is a great option for:
    - Organizations requiring in-house model tuning and full data control
    - Use cases like document summarization, code generation, classification, predictive pipelines
- Mistral was by far my favorite to run of the LLMs I had chosen. It's ability to interpret the prompt and respond, the accuracy of the data, and the interaction was the easiest to run and apply.

Flan T-5

Baseline

- Developed by Google Research, openly available via Hugging Face
- Instruction-tuned T5 (Text-To-Text Transfer Transformer)
- Flan version is trained on over 1,000 tasks with task-specific prompts
- Supports Base (250M), XL (3B), and XXL (11B) parameter sizes
- Apache 2.0 License is free for commercial use

Performance and Setup

- Tested google/flan-t5-Base using Google Colab
- Colab GPU Runtime T4 used for inference
- Runtime: ~2–3 seconds for full response
- Response results:
    - Clear and concise with general summary
    - Struggled more with data heavy tasks
        - Took multiple prompt change attempts to get through
        - Did eventually receive meaningful results
    - Simple code results performed well

Cost

- Can run locally or on cloud GPUs
- Inference cost (Colab): ~$0.01–$0.10 per run (cost of GPU time)
- Fine-tuning with LoRA possible (using PEFT library)
- On-prem setup with RTX 4090 (~$1,800) supports XL
- No API/tokens — full offline access

Pros/Cons

- Strengths:
  - Structured answers
  - technical fluency
  - business-usable outputs
  - Excellent for document processing, internal automation, or quick plans
- Limitations:
  - No multimodal input (text-only)
  - ~512 token context limit
  - No “chat-style” interactivity (no retained memory across prompts)
- Performance on par with larger closed models (e.g. GPT-3) in structured tasks

Summary

- Best for orgs needing fully local LLMs
- Great for secure & private deployments
- Use LongT5 or chunking for long docs
- Not designed for image input (use BLIP2 + Flan-T5 for that)
- Use cases: structured generation, internal copilots

Falcon

Baseline

- Open-source LLM series by Technology Innovation Institute (UAE)
- Models: 7B, 40B (Apache 2.0); 180B (TII Research License)
- Trained on curated web data (RefinedWeb) — up to 3.5 trillion tokens
- Optimized for high-throughput inference & instruction following

Performance and Setup

- Used tiiuae/falcon-7b-instruct via Hugging Face Transformers
- Inference on Google Colab GPU T4
- Prompted similarly to Mistral/Flan-T5: predictive maintenance scenario
- Output was clear, step-wise, and technically detailed
- Inference: ~2–4 seconds per full response

Cost

- Falcon 7B can run on single RTX 3090/4090 GPU
- Cloud (A100): ~$1.25/hour for 40B; ~$8–10/hour for 180B
- Fine-tuning with QLoRA or PEFT recommended for lower VRAM
- No per-token cost — local inference possible with full control
- Falcon 40B/180B suited for enterprise-scale RAG, summarization, QA

Pros/Cons

- Strengths:
  - High-quality completions
  - instruction-ready
  - low latency
- Falcon 7B has a good balance of size & performance
- Limitations:
  - No native support for image input
  - Instruction finetuning weaker than Mistral or ChatGPT
  - Falcon 180B needs cluster-scale hardware

Summary

- Great for teams wanting scalable open LLMs
- Falcon 7B = cost-effective, easy to run
- Falcon 40B = sweet spot for performance in enterprise
- Falcon 180B = top-tier benchmark performance (MMLU ~68%)
- No vendor lock-in, strong license (Apache 2.0 for 7B/40B)

Grok

Baseline

- **Baseline Model:** Grok
- **Use Case Focus:** General-purpose conversational AI, designed to engage users with informative and contextually aware dialogue
- **Ideal Fit For:** Casual users and general-purpose AI interaction within a consumer platform
- **Key Takeaway:** Grok provides a playful, personality-driven chat experience directly inside X, offering free access but lacking transparency, customization, and offline deployment options.

Performance and Setup

- **Memory Footprint:** Not applicable - model is entirely cloud-based
- **Runtime:**
  - Cold boot: Not exposed to user
  - Inference: ~3-5 seconds per prompt
- **Soft Cost:** No setup required; runs in browser on X

Cost

- **Token cost:** $3.00 per million input tokens
- **Access:** Free version integrated into X Premium, but with limited functionality compared to the super version

Deployment Type	Description	Cost (Hard)	Setup Complexity
SuperGrok	Enhanced access and speed	$30/month	Low

Pros and Cons

- **Supports:**
  - Conversational inference on general-purpose, natural language prompts
  - Web-based access through X with no local setup required
  - Real-time search (DeepSearch) available in premium tiers
- **Does Not Support:**
  - Local or offline deployment — no self-hosted model access
  - Customization or fine-tuning
  - Integration with external APIs or developer tools
- **Security Considerations:**
  - All interactions hosted via X infrastructure — no user control over model environment
  - User queries are linked to X accounts and may be logged
  - No transparency into training data or data retention practices
  - Not suitable for use cases requiring data confidentiality or regulatory compliance

Summary

- Grok is a great option for:
  - Users seeking a conversational assistant integrated directly into the X platform
  - Casual or general-purpose language generation without the need for setup or infrastructure
  - Teams or individuals interested in experimenting with AI-generated commentary, and creative outputs
  - Organizations exploring the integration of AI into social media platforms or consumer-facing web interfaces without managing infrastructure

OpenLlama

Baseline

- **Baseline Model:** OpenLLaMA 3B, 7B, and 13B
- **Use Case Focus:** Open-source LLM research and custom deployments
- **Tested On:** Google Colab
- **Ideal Fit For:** Organizations seeking a fully customizable, license-free LLM for experimentation and fine-tuning
- **Key Takeaway:** OpenLLaMA is an open, scalable alternative to Meta's LLaMA. Enables research and enterprise AI development with full model control.
    - Good for companies who want to customize their AI tools

Performance and Setup

- **Initial Setup:** ~13.5 GB for the 7Bv2 version
- **Runtime:**
  - Inference: ~1-3 seconds per prompt on a capable GPU
- **Runtime (Fine-Tuning):** Requires multi-hour sessions depending on dataset and hardware. Best results when using Hugging Face Trainer or EasyLM.
- **Soft Cost:** ~3-4 hours to set up environment, install packages, and handle tokenizer setup, and ~1-2 minutes to load model, ~1-2 minutes per reset in notebook environment

Cost

- **Token cost:** None
  - Free under Apache 2.0 license
- **Access:** No gated access required - weights available from Hugging Face/GitHub without approval

Deployment Type	Description	Cost (Hard)	Setup Complexity
Colab Pro+ (GPU)	Run smaller models (3B or 7B)	$49/month	Low
Paperspace	Jupyter style interface	~$0.50-$2/hr	Medium

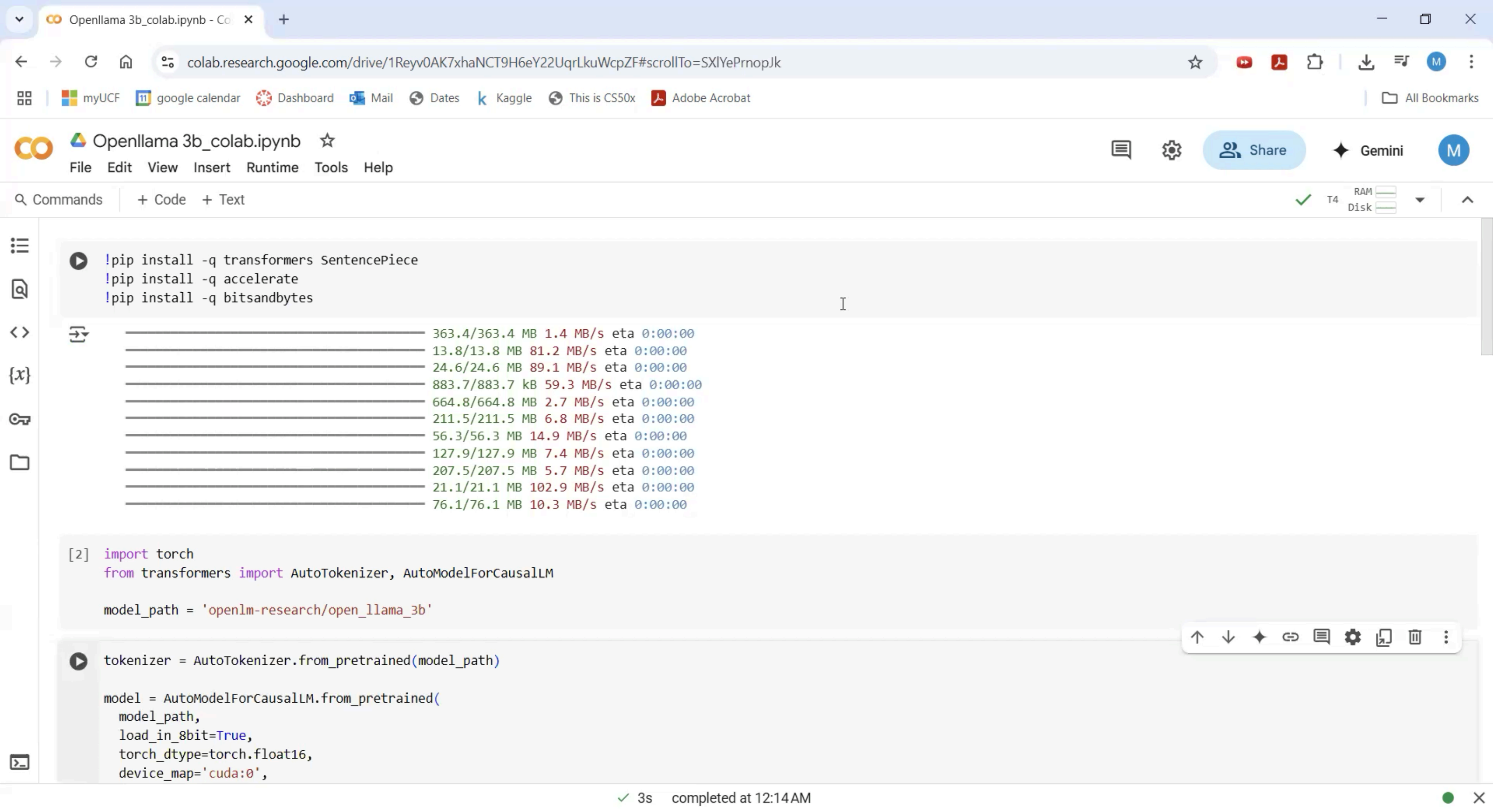
Pros and Cons

- **Supports:**
  - Inference on structured and natural language prompts
  - Drop-in compatibility with LLaMA implementations and tokenizer architecture
  - Integration with Hugging Face Transformers and EasyLM
  - Self-hosted deployment on local hardware or cloud (AWS, GCP, Colab, etc.)
- **Does Not Support:**
  - Instruction-following or chat-based responses (without fine-tuning)
  - Multimodal input (e.g., images, PDFs, audio)
  - Real-time web access or retrieval-augmented generation (RAG)
  - Memory or session history for multi-turn conversations
- **Security Considerations:**
  - Fully self-contained and open-source — no third-party API dependency
  - No telemetry or usage tracking — suitable for sensitive internal deployments

Summary

- OpenLlama is a great option for:
  - Teams or researchers needing full control over how the model is used and trained
  - Use cases requiring on-premises or private cloud deployment of large language models
  - Teams looking to fine-tune language models using domain-specific datasets
  - Applications involving secure document summarization, virtual assistants, internal analytics, or custom NLP pipelines
  - Developers requiring open licensing (Apache 2.0) for commercial use without vendor lock-in

Demo



```
!pip install -q transformers SentencePiece
!pip install -q accelerate
!pip install -q bitsandbytes
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 1.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 81.2 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 89.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 59.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 6.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 14.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 7.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 5.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 102.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.1/76.1 MB 10.3 MB/s eta 0:00:00
```

```
[2] import torch
    from transformers import AutoTokenizer, AutoModelForCausalLM

    model_path = 'openlm-research/open_llama_3b'
```

```
tokenizer = AutoTokenizer.from_pretrained(model_path)

model = AutoModelForCausalLM.from_pretrained(
  model_path,
  load_in_8bit=True,
  torch_dtype=torch.float16,
  device_map='cuda:0',
```

R1 1776

Baseline

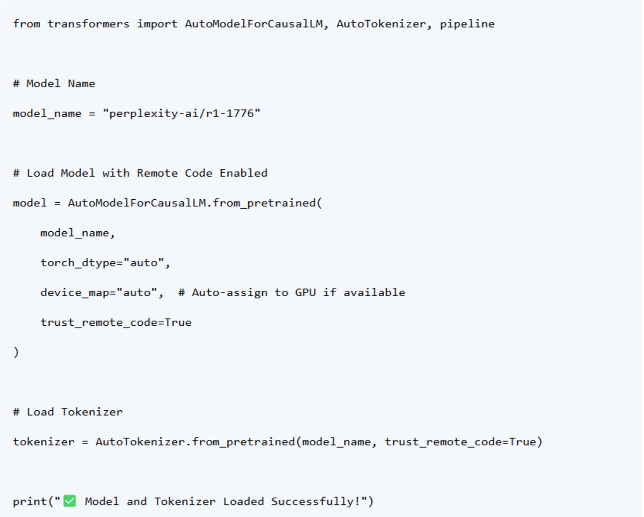
- **Baseline Model:** R1
- **Use Case Focus:** Open-source LLM focusing on an unbiased response model
- **Tested On:** Perplexity AI, Jupyter Notebook
- **Ideal Fit For:** Organizations looking for factual integrity without the risk of undocumented data filtration
- **Key Takeaway:** Released by Perplexity AI, its model weights and documentation are publicly accessible on Hugging Face. Can be used via API or locally.
  - Unconstrained by geopolitical restrictions
  - Uncensored data that can be used in the context of International Events

Performance and Setup, Perplexity Pro

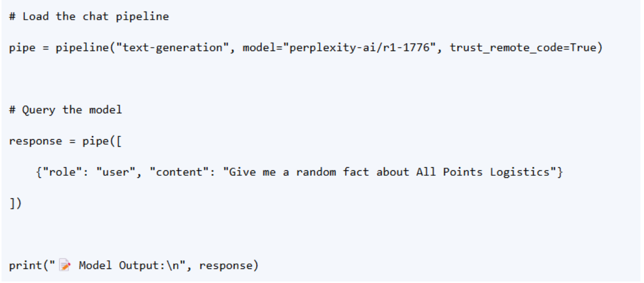
- **Setup:** For Cloud version, simply selecting R1 1776 within Perplexity. To run locally, it must be run through Perplexity Labs.
- **Runtime:**
  - Inference: ~7-9 seconds per prompt
  - Greatly affected by the complexity of the prompt
- **Soft Cost:** If running through Local/API, ~1 minute to load relevant files, ~1-2 minutes to formulate prompts

Performance and Setup, Local Usage

- **Setup:** Local Version
- **Minimum Specifications:**
  - **Single GPU Setup** (May require model offloading to CPU)
  - **GPU:** 1x NVIDIA A100 80GB / RTX 6000 Ada 48GB (Offloading required)
  - **vRAM: 48GB+**
  - **CPU:** 16 vCPUs (Recommended)
  - **RAM: 64GB+**
  - **Storage: 200GB+ NVMe SSD**
  - **Framework:** PyTorch + transformers (Accelerate enabled)
  - **Torch Precision:** bfloat16 / float16
- **Runtime:**
  - Inference: ~7-9 seconds per prompt
  - Greatly affected by the complexity of the prompt



```
from transformers import AutoModelForCausalLM, AutoTokenizer, pipeline


# Model Name

model_name = "perplexity-ai/r1-1776"


# Load Model with Remote Code Enabled

model = AutoModelForCausalLM.from_pretrained(

    model_name,

    torch_dtype="auto",

    device_map="auto",  # Auto-assign to GPU if available

    trust_remote_code=True

)


# Load Tokenizer

tokenizer = AutoTokenizer.from_pretrained(model_name, trust_remote_code=True)


print("✅ Model and Tokenizer Loaded Successfully!")
```



```
# Load the chat pipeline

pipe = pipeline("text-generation", model="perplexity-ai/r1-1776", trust_remote_code=True)


# Query the model

response = pipe([

    {"role": "user", "content": "Give me a random fact about All Points Logistics"}

])


print("📝 Model Output:\n", response)
```

- **Soft Cost:** If running Jupyter Notebook/similar, ~2 hours to load model, ~1 minute to load relevant files, ~1-2 minutes to formulate prompts

Cost

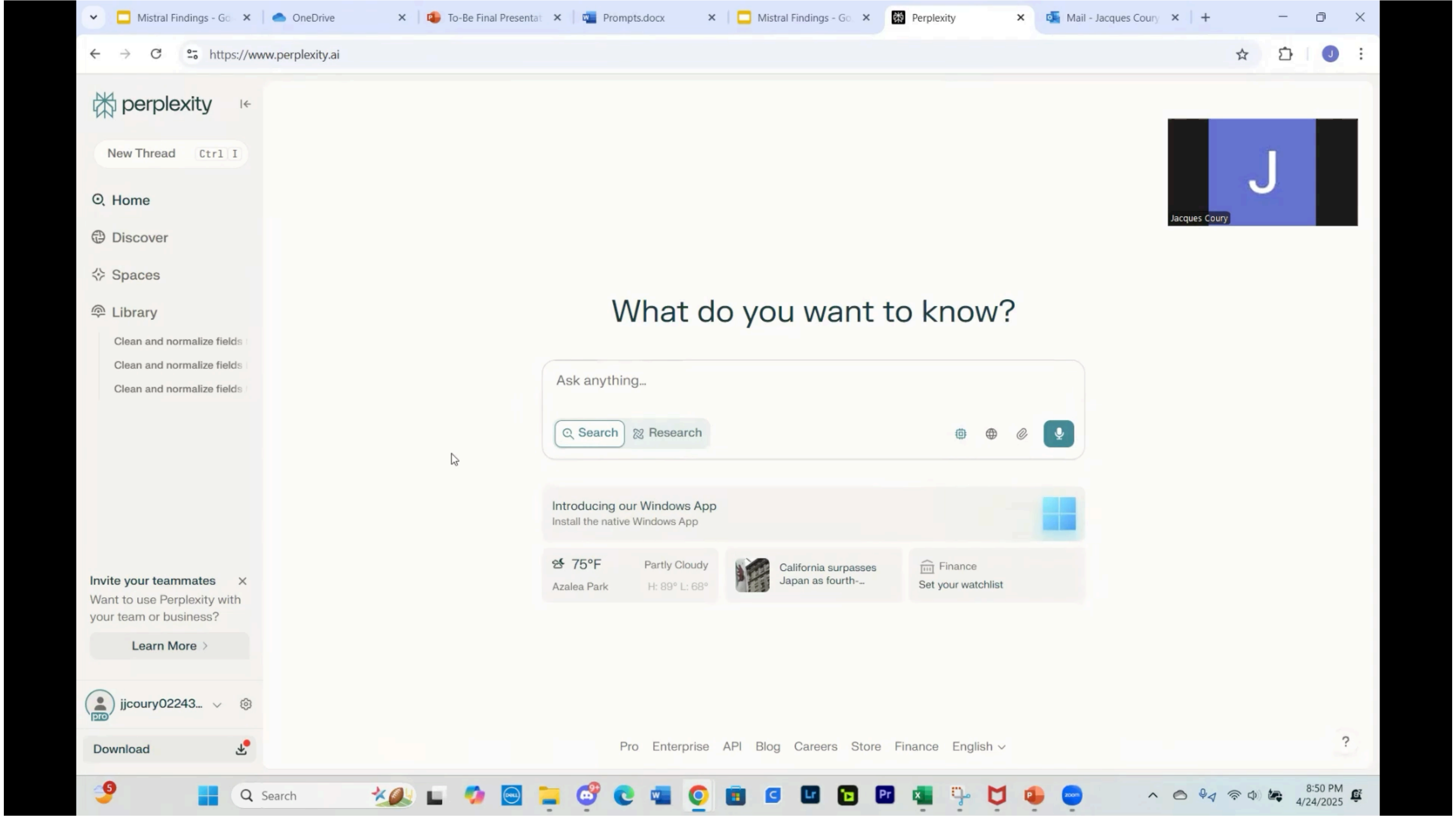
- **Self-Setup:** Requires more labor hours, but more customizability
  - *Deciding factor would be to consider would be the need to modify existing LLM*
- **Pro Search:** "Pro Search is an advanced feature that goes beyond our free Auto search option. It conducts thorough research to provide in-depth, accurate responses to your questions".
  - Also tends to ask follow up questions to prompt to ensure project scope is being met
  - Performs simulations and runs data analysis

Deployment Type	Description	Cost (Hard)	Setup Complexity
Jupyter Notebook	No cost, not as UI friendly Can be freely modified/	No Hard Cost	Medium
Perplexity Pro	300+ Pro searches Unlimited files Usage of multiple AIs User Friendly Follow up prompts	$20/mo	Low

Pros/Cons/Summary

- **Benefits:**
  - **Pro Version:** Extremely user-friendly UI
  - **Pro Version:** Relatively low cost when comparing to the potential labor set up time of more complex LLMs
  - **Pro Version:** Follow up questions are given after prompt completion relating to solving your original query
  - **Locally:** No cost
  - **Locally:** Can be modified to tailor fit to Company's needs
  - **Both:** Completely uncensored when dealing with International Topics
- **Drawbacks:**
  - Large scale implementation almost guarantees need for Pro version (added hard cost)
  - Not ideal for overly complex searches
  - Response time varies greatly depending on search, higher wait times when compared to other LLMs
  - Memory or session history for multi-turn conversations
- **Security Considerations:**
  - The ability to be freely modify may lead to gateways of misuse (for sensitive content) when using cloud.
  - Running model locally allows for control over access permissions, and no API data leaks.

Demo (Perplexity Pro)



Tabby ML

Baseline

- **Baseline Model:** Tabby ML
- **Use Case & Ideal Fit:** Companies looking for local and secure control with a coding and data LLM.
- **Tested On:** Docker
- **Key Takeaway:** Tabby ML is a great alternative to any sort of cloud-based data/ coding help
  - Local inference aides in Security Concerns
  - Virtually no cost due to local setup and running

Performance and Setup

- **Setup:** ~1.5hrs with Docker
- **Runtime:**
  - Cold Start: ~30-45 seconds
  - Inference: ~3 seconds for standard code prompts=
- **Fine Tuning:** Pre-trained models available on Hugging Face

Cost

- Can run locally or on cloud GPUs
- Inference cost on Colab/Local (Docker): ~$0.01–$0.10/hour

Pros/Cons/Summary

- **Benefits:**
  - Docker container provides very easy routes of deployment/usage
  - Local inference provides security
  - Air gapped environment availability
  - Alternative to CoPilot
- **Drawbacks:**
  - Does not support Multimodal/image inputs, mainly limited to coding help
  - No access to prompt history/memory across sessions
- **Security Considerations:**
  - As previously stated, best suited for environments in which sensitive data or proprietary information it being utilized

Final Recommendations

- Shraavani's Best
  - Claude 2.1
- Jacob's Best
  - M365 Copilot
- Maria's Best
  - OpenLlama 7Bv2
- Jacques' Best
  - Tabby ML
- Kevin's Best
  - Mistral

Deployment Recommendations

Primary Recommendations:

1. **Implement Control Plan:** Actively deploy and utilize the monitoring and control mechanisms (KPIs, reviews, response plans)
    1. Prioritize establishing secure cloud environments (GCP Vertex AI FedRAMP / M365 GCC High) and finalizing internal LLM usage policies immediately.
    2. Conduct controlled pilots for both selected LLM (initial rollout) and the specialized API (e.g., for Launch Support analysis), focusing on metrics, user feedback, and Go/No-Go criteria.
2. **Invest in Change Management:** Commit resources to comprehensive user training (including responsible AI and compliance) and ongoing support.
3. **Establish AI Governance:** Form an internal body to oversee ongoing LLM use, risk, compliance, and strategy.

Project Deliverables Summary

- **Key Deliverables Provided to APL & UCF:**
  - Lean Six Sigma Project Charter
  - Project Proposal Report
  - AS-IS Analysis Report
  - Final TO-BE Analysis Report
  - Final Presentation
  - Supporting Appendices

Opportunities for Future Work

- **Implementation:** Full execution of the recommended Deployment Roadmap
- **Integration:** Deeper technical integration with core APL systems
- **Expansion:** Systematic evaluation of LLMs for additional APL capability areas
- **Advancement:** Exploration of secure fine-tuning, LLM agents for automation, and advanced multimodal applications as technology and APL maturity evolve.
- **Governance:** Continuous refinement and execution of the AI Governance framework and Control Plan.

Questions & Discussion

Thank You to All Points Logistics for their sponsorship & collaboration!

Thank You to all UCF Faculty and Mentors!

Thank You to everyone else!