**1. Core AI and ML Concepts**

* **Artificial Intelligence (AI):** Refers to building systems that mimic human intelligence—used across Google services like Google Assistant, Translate, and Search.
* **Machine Learning (ML):** A subset of AI where systems learn from data. Google Cloud AI Platform and TensorFlow are key tools here.
* **Generative AI:** Focuses on creating new content (text, images, etc.). Google’s Bard and Imagen are examples.
* **Deep Learning:** Uses neural networks with many layers—central to Google’s speech recognition, image classification, and LLMs.
* **Foundation Models:** Large-scale models trained on vast data, such as Google’s PaLM (Pathways Language Model).
* **Large Language Models (LLMs):** Specialized in understanding and generating human language—used in Google Search, Docs (smart compose), and Bard.

**2. Data and Its Role in AI**

* **Structured Data:** Organized data (e.g., spreadsheets, SQL databases)—used in Google BigQuery and Cloud SQL.
* **Unstructured Data:** Includes text, images, audio—processed using Google Cloud Vision, Natural Language API, and Speech-to-Text.
* **Data Quality:** Emphasized for accurate model training.
* **Data Accessibility:** Ensures data is usable and in the right format—Google Cloud Storage and DataPrep help with this.

**3. ML Lifecycle (Aligned with Google Cloud AI Platform)**

* **Data Ingestion & Preparation:** Google Cloud Dataflow, Dataprep, and BigQuery support this phase.
* **Model Training:** Google Vertex AI and TensorFlow are used to train models.
* **Model Deployment:** Vertex AI and AI Platform Prediction allow models to be served at scale.
* **Model Management:** Tools like Vertex AI Model Monitoring help track model performance over time.

**4. Responsible AI (Aligned with Google’s AI Principles)**

* **Secure AI:** Protects models from misuse—Google’s Secure AI Framework (SAIF) is referenced.
* **Ethical AI:** Ensures fairness, transparency, and accountability—Google’s AI Principles guide this.

**5. Learning Approaches in ML**

* **Supervised Learning:** Used in Google Photos (face recognition), Gmail (spam detection).
* **Unsupervised Learning:** Powers Google News clustering and YouTube recommendations.
* **Reinforcement Learning:** Applied in robotics and game-playing AI (e.g., DeepMind’s AlphaGo).

**Google Cloud Platform: Vertex AI**

**Vertex AI** is Google Cloud’s unified ML platform that supports the full ML lifecycle:

* **Model Garden**: A repository of pre-trained models from Google, third parties, and open-source communities.
* **Model Builder**:
  + **Custom Models**: Train models from scratch using ML frameworks.
  + **AutoML**: Build models with minimal technical expertise.

Vertex AI simplifies:

* Model training
* Deployment
* Management of ML and Gen AI solutions

**Infrastructure**

The foundational layer includes:

* **Hardware**: Servers, GPUs, TPUs
* **Software**: Tools to train, store, and run AI models

**AI on the Edge**

Running AI closer to the data source or user:

* **Lite Runtime (LiteRT)**: Google’s tool for deploying AI models on edge devices.
* **Gemini Nano**: A compact, efficient AI model optimized for on-device performance.

**Gemini Tooling for Personal Productivity**

**a. Gemini App**

* A generative AI chatbot designed to assist with:
  + Writing
  + Planning
  + Learning
  + General productivity tasks

**b. Gemini for Google Workspace**

* Integrates generative AI into familiar Google Workspace apps:
  + **Gmail**: Compose emails using AI assistance.
  + **Slides**: Generate images directly within presentations.
  + **Meet**: Summarize meeting notes automatically.

**c. Gemini for Google Cloud**

* Acts as an AI assistant for cloud developers and administrators:
  + Write and debug code
  + Manage and optimize cloud applications
  + Analyze data in **BigQuery**
  + Strengthen security posture

**d. NotebookLM**

* A research assistant tool that:
  + Allows file uploads
  + Summarizes key points
  + Answers questions
  + Generates ideas
  + Maintains grounding in the uploaded source material

**Streamlining Prompting Workflows**

Techniques to enhance productivity and consistency when interacting with AI:

* **Reusing Prompts**: Save prompts as templates for repeated use.
* **Prompt Chaining**: Maintain context by continuing conversations within the same chatbot session.
* **Saved Info in Gemini**: Store specific information for consistent reuse by the model.
* **Gems**: Personalized AI assistants within Gemini that:
  + Follow specific instructions
  + Streamline workflows
  + Use templates and guided interactions

**Prompting Techniques**

Different strategies to guide the AI’s behavior and output:

* **Zero-shot**: No examples provided; the model infers the task from the prompt alone.
* **One-shot**: One example is given to guide the model.
* **Few-shot**: Multiple examples are provided to improve accuracy.
* **Role Assignment**: Assign a persona to the model to influence tone, style, and focus.
* **Prompt Chaining**: Engage in iterative dialogue to refine responses.

**Model Guidance and Refinement**

Techniques to ensure the AI’s output is accurate and trustworthy:

* **Grounding**: Ensures AI responses are connected to verifiable sources.
* **RAG (Retrieval-Augmented Generation)**:
  1. **Retrieving**: The model pulls relevant information from a large knowledge base.
  2. **Generating**: It then uses this information to produce a grounded and accurate response.

| **Technique** | **When to Use It** |
| --- | --- |
| **Zero-shot** | When no examples are needed. E.g., “Summarize this article.” |
| **One-shot** | When one example helps clarify the task. |
| **Few-shot** | When multiple examples improve accuracy. |
| **Role Prompting** | When tone or perspective matters. E.g., “As a project manager, write a status update.” |
| **Prompt Chaining** | When refining or continuing a conversation with the AI. |

**Gemini Ecosystem**

| **Concept** | **Practical Example** |
| --- | --- |
| **Gemini App** | You’re writing a project proposal. You ask Gemini: “Draft a proposal for a new quality assurance tool for our engineering team.” |
| **Gemini for Google Workspace** | In Gmail, you type “Summarize this customer complaint and draft a polite response.” Gemini generates a reply. In Slides, you say “Create an image of a QA workflow” and it inserts a visual. |
| **Gemini for Google Cloud** | You’re deploying a new microservice. You ask Gemini: “Generate a Cloud Run deployment YAML for a Node.js app with 512MB memory.” |
| **NotebookLM** | You upload a test plan document and ask: “Summarize the key test cases and identify any missing coverage areas.” |

| **Technique** | **Description** | **Example** |
| --- | --- | --- |
| **Zero-shot** | No examples | “Summarize this article.” |
| **One-shot** | One example | “Here’s one test case. Write another.” |
| **Few-shot** | Multiple examples | “Here are 3 bug reports. Write a 4th.” |
| **Role Prompting** | Assign a persona | “As a senior engineer, write a review.” |
| **Prompt Chaining** | Iterative refinement | “Generate a report → Add summary → Add recommendations.” |

**Customer Engagement Suite (Built on Google CCaaS)**

This suite is designed to enhance customer interaction and support, leveraging cloud-native tools:

* **Conversational Agents**: AI-powered chatbots that interact with customers using natural language.
* **Agent Assist**: Provides real-time support to human agents by suggesting responses and surfacing relevant information.
* **Conversational Insights**: Analyzes customer interactions to extract actionable insights and trends.

These tools are part of Google’s **Contact Center as a Service (CCaaS)** platform, which is enterprise-grade and cloud-native.

**Vertex AI Search**

A powerful tool for implementing search and recommendation systems within your business. It enables:

* Semantic search capabilities.
* Personalized recommendations.
* Integration with internal data sources.

**Agentspace**

A framework to embed intelligent agents into internal dashboards or websites. These agents can:

* Access and understand data from various internal sources.
* Provide conversational interfaces for internal users.

**RAG: Retrieval-Augmented Generation**

A key architecture for enhancing LLM responses with external data:

1. **Retrieval**: Pulls relevant information from external sources.
2. **Augmentation**: Injects retrieved data into the prompt.
3. **Generation**: LLM generates a response based on the enriched prompt.
4. **Iteration (optional)**: The process can repeat to refine the response.

**Types of Agents**

* **Deterministic Agents**: Follow predefined rules and paths (traditional approach).
* **Generative Agents**: Use LLMs and natural language to create dynamic, conversational experiences.

**Reasoning Loop & Prompt Engineering Techniques**

These techniques enhance the reasoning capabilities of LLMs:

* **ReAct (Reason and Act)**: Combines reasoning with actions to solve user queries.
* **CoT (Chain-of-Thought)**: Breaks down complex problems into intermediate reasoning steps.
* **Metaprompting**: Uses prompts to generate or modify other prompts, guiding the model’s behavior.

**Tooling for Gen AI Agents**

* **Extensions**: Connect agents to external APIs.
* **Functions**: Define specific tasks the agent can perform.
* **Data Stores**: Provide structured access to information.
* **Plugins**: Add new capabilities or integrations to the agent.

**Types of Agents**

| **Type** | **Description** | **Use Case** |
| --- | --- | --- |
| **Deterministic** | Rule-based, predefined logic | IVR systems, form-based bots |
| **Generative** | LLM-powered, natural language | AI chatbots, virtual assistants |

**Reasoning Loop & Prompt Engineering Techniques**

| **Technique** | **Description** |
| --- | --- |
| **ReAct (Reason and Act)** | Combines reasoning with actions to solve queries |
| **CoT (Chain-of-Thought)** | Breaks down problems into intermediate reasoning steps |
| **Metaprompting** | Prompts that generate or modify other prompts |

Gemini

Trained on a massive dataset of text, images, code, audio, video, and more. This multimodal training allows Gemini to perform tasks across various domains, including language understanding, image generation, code generation, and more.

Chirp

Trained on a large dataset of audio in various languages. Chirp is designed for speech recognition and can be used for tasks like voice assistants, transcription, and translation. Find information at the right time.

Imagen

Trained primarily on a massive dataset of images and text descriptions. This enables Imagen to generate high-quality images from text descriptions, perform image editing tasks, and understand the content of images.

Google introduces Gemini, a multimodal AI model capable of understanding and integrating various information types. The following year, Google expands the Gemini ecosystem with Gemini 1.5, bringing it to more products and launching Gemini Advanced, which provides access to Google's most capable AI models.

**2024: Gemma**

Google announces Gemma, a family of lightweight state-of-the-art open models built from the same research and technology used to create the Gemini models.

Gen AI has been built into tooling across Google’s comprehensive ecosystem to support your organization in their day-to-day work. Here are some examples:

* **Google Search:** Leverage the power of Gemini in Google Search to find information faster and with more accuracy.
* **Gemini for Google Workspace:** Draft emails in Gmail, generate presentations in Slides, summarize meeting notes in Docs, and automate tasks in Sheets with integrated gen AI features.
* **Gemini App:** Access the power of Gemini directly through a dedicated app for personalized assistance and creative exploration.
* **Gemini for Google Cloud:** Build applications and services on Google Cloud using powerful Gemini models.

Using gen AI responsibly, ethically, and securely is extremely important for all organizations. Google has a commitment to building AI responsibly by building an ecosystem that is built securely and ethically and, at the same time, providing you educational resources for doing the same as you build your own solutions. Here are some examples:

* **Secure AI Framework (SAIF):** A comprehensive suite of tools and best practices for building secure AI systems.
* **Mandiant:** Leverage Mandiant's threat intelligence and expertise to protect your AI systems from cyberattacks.
* **AI Principles:** Google's published AI Principles guide the development and deployment of AI technologies.
* **Responsible AI Toolkit:** Resources and tools to help developers and organizations build AI systems that are fair, unbiased, and socially beneficial.

Building gen AI for an enterprise involves thinking about security measures, data governance, compliance certifications, and more. Through using Google Cloud’s ecosystem, you know you are building off an enterprise-ready system. Here are some examples:

* **Vertex AI:** A unified platform for building and deploying machine learning models, including Gemini. It provides enterprise-grade security, scalability, and compliance.
* **Google Cloud's security infrastructure:** Benefit from Google's world-class security infrastructure, including data encryption, access control, and network security.
* **Compliance certifications:** Google Cloud offers services that support compliance with a wide range of industry standards and regulations, including ISO 27001, SOC 2, and HIPAA.

Google has an open approach so you are not stuck to any one solution and you have the flexibility to choose what works best for your organization. Here are some examples:

* **Contributions to TensorFlow and PyTorch:** Google actively contributes to popular open source machine learning frameworks.
* **Open models and datasets:** Google releases pre-trained models and datasets to the research community.
* **Support for open standards:** Google supports open standards for AI interoperability and data exchange.

**Labeled data** has tags, such as a name, type, or number. These tags, whether applied manually or by automated systems, assign meaning to the data.

For instance, an image dataset for training a cat-detection model would label each picture as either a cat or dog. Similarly, a set of customer reviews might be labeled as positive, negative, or neutral. These labels enable algorithms to learn relationships and make accurate predictions.

**Unlabeled data** is simply data that is not tagged or labeled in any way. It's raw, unprocessed information without inherent meaning.

Examples of unlabeled data include a collection of unorganized photos, a stream of audio recordings, or website traffic logs without user categorization. In these cases, algorithms must independently discover patterns and structures within the data, as there are no pre-existing labels to guide the learning process.

**Supervised machine learning** trains models on labeled data, where each input is paired with its correct output, allowing the model to learn the relationship between them. The model's goal is to identify patterns and relationships within this labeled data, enabling it to accurately predict outputs for new, unseen inputs.

Predicting housing prices is a common example of supervised learning. A model is trained on a dataset where each house has labeled data, such as its size, number of bedrooms, location, and the corresponding sale price. This labeled data allows the algorithm to learn the relationship between the features of a house and its price. Once trained, the model can then predict the price of a new house based on its features.

**Unsupervised ML models** deal with raw, unlabeled data to find natural groupings. Instead of learning from labeled data, it dives headfirst into a sea of unlabeled data.

For example, an unsupervised learning algorithm could analyze customer purchase history from your company's database. It might uncover hidden segments of customers with similar buying habits, even though you never explicitly labeled those segments beforehand. This can be incredibly valuable for targeted marketing or product recommendations.

Think of it as exploratory analysis. Unsupervised learning helps you understand the underlying structure of your data and uncover insights that you might not have even known to look for.

**Reinforcement learning** is all about learning through interaction and feedback. Imagine a robot learning to navigate a maze. It starts with no knowledge of the maze's layout. As it explores and interacts with the maze, it collects data—bumping into walls (negative feedback) or finding shortcuts (positive feedback). Through this process of trial and error, the algorithm learns which actions lead to the best outcomes.

Google Cloud supports data ingestion through several tools.

* **Pub/Sub** handles real-time streaming data processing, regardless of the structure of the data.
* **Cloud Storage** is well-suited for storing unstructured data.
* **Cloud SQL** and **Cloud Spanner** are used to manage structured data.

Google Cloud offers tools like **BigQuery** for data analysis and **Data Catalog** for data governance. These tools help prepare data for ML models.

With BigQuery, you can filter data, correct its inconsistencies, and handle missing values.

With Data Catalog, you can find relevant data for your ML projects. This tool provides a centralized repository to easily discover datasets in your organization.

The process of creating your ML model using data is called model training. **Google Cloud's Vertex AI platform** provides a managed environment for training ML models.

With Vertex AI, you can set parameters and build your model, using prebuilt containers for popular machine learning frameworks, custom training jobs, and tools for model evaluation. Vertex AI also provides access to powerful computing resources to make the model training process faster.

Model deployment is the process of making a trained model available for use.

**Vertex AI**simplifies this by providing tools to put the model into action for generating predictions. This includes scaling the deployment, which means adjusting the resources allocated to the model to handle varying amounts of usage.

Model management is the process of managing and maintaining your models over time. Google Cloud offers tools for managing the entire lifecycle of ML models. This includes the following:

* **Versioning**: Keep track of different versions of the model.
* **Performance tracking**: Review the model metrics to check the model's performance.
* **Drift monitoring**: Watch for changes in the model's accuracy over time.
* **Data management**: Use Vertex AI Feature Store to manage the data features the model uses.
* **Storage**: Use Vertex AI Model Garden to store and organize the models in one place.
* **Automate**: Use Vertex AI Pipelines to automate your machine learning tasks.

[**Modality**](https://storage.googleapis.com/cloud-training/cls-html5-courses/C-GENDEC-B/v1.0.0/C-GENDEC-B%20-%20Gen%20AI%20Unlock%20Foundational%20Concepts/index.html)

When selecting a generative AI model, it's crucial to consider the modality of your input and output. Modality refers to the type of data the model can process and generate, such as text, images, video, or audio. If your application focuses on a single data type, like generating text-based articles or creating audio files, you'll want to choose a model optimized for that specific modality. For applications that require handling multiple data types, such as generating image captions (processing images and producing text) or creating video with accompanying audio, you'll need a multimodal model. These models can understand and synthesize information across different modalities

[**Context window**](https://storage.googleapis.com/cloud-training/cls-html5-courses/C-GENDEC-B/v1.0.0/C-GENDEC-B%20-%20Gen%20AI%20Unlock%20Foundational%20Concepts/index.html)

The context window refers to the amount of information a model can consider at one time when generating a response. A larger context window allows the model to "remember" more of the conversation or document, leading to more coherent and relevant outputs, especially for longer texts or complex tasks. However, larger context windows often come with increased computational costs. You need to balance the need for context with the practical limitations of your resources.

[**Security**](https://storage.googleapis.com/cloud-training/cls-html5-courses/C-GENDEC-B/v1.0.0/C-GENDEC-B%20-%20Gen%20AI%20Unlock%20Foundational%20Concepts/index.html)

Security is paramount, especially when dealing with sensitive data. Consider the model's security features, including data encryption, access controls, and vulnerability management. Ensure the model complies with relevant security standards and regulations for your industry.

[**Availability and reliability**](https://storage.googleapis.com/cloud-training/cls-html5-courses/C-GENDEC-B/v1.0.0/C-GENDEC-B%20-%20Gen%20AI%20Unlock%20Foundational%20Concepts/index.html)

The availability and reliability of the model are crucial for production applications. Choose a model that is consistently available and performs reliably under load. Consider factors like uptime guarantees, redundancy, and disaster recovery mechanisms.

**Gemini**

Gemini, a multimodal model, can understand and operate across diverse data formats, such as text, images, audio, and video. Gemini's multimodal design supports applications that require complex multimodal understanding, advanced conversational AI, content creation, and nuanced question answering.

**Gemma**

A family of lightweight, open models is built upon the research and technology behind Gemini. They offer developers a user-friendly and customizable solution for local deployments and specialized AI applications.

**Imagen**

A powerful text-to-image diffusion model, it excels at generating high-quality images from textual descriptions. This makes it invaluable for creative design, ecommerce visualization, and content creation.

**Veo**

A model capable of generating video content. It can produce videos based on textual descriptions or still images. Its functionality allows for the creation of moving images for applications such as film production, advertising, and online content.

| **Feature** | **RAG** | **Fine-tuning** | **Grounding** |
| --- | --- | --- | --- |
| **Definition** | **Augments LLMs by retrieving relevant information from external knowledge bases and adding it to the prompt.** | **Further trains a pre-trained model on a new dataset to adapt it to a specific task or domain.** | **Connects an AI model's output to verifiable sources of information.** |
| **Process** | **Retrieve relevant information. → Add it to the prompt. → Generate a response.** | **Select a pre-trained model. → Gather data. → Train the model. → Evaluate and refine.** | **Provide access to data sources. → Use RAG or fine-tuning to connect the output.** |
| **Data sources** | **External knowledge bases (databases, documents, internet).** | **Task- or domain-specific datasets.** | **External knowledge bases or specific datasets.** |
| **Relationship to grounding** | **A specific technique for achieving grounding.** | **Improves a model's ability to be grounded in specific domains.** | **The overarching goal, achieved through techniques like RAG and fine-tuning.** |