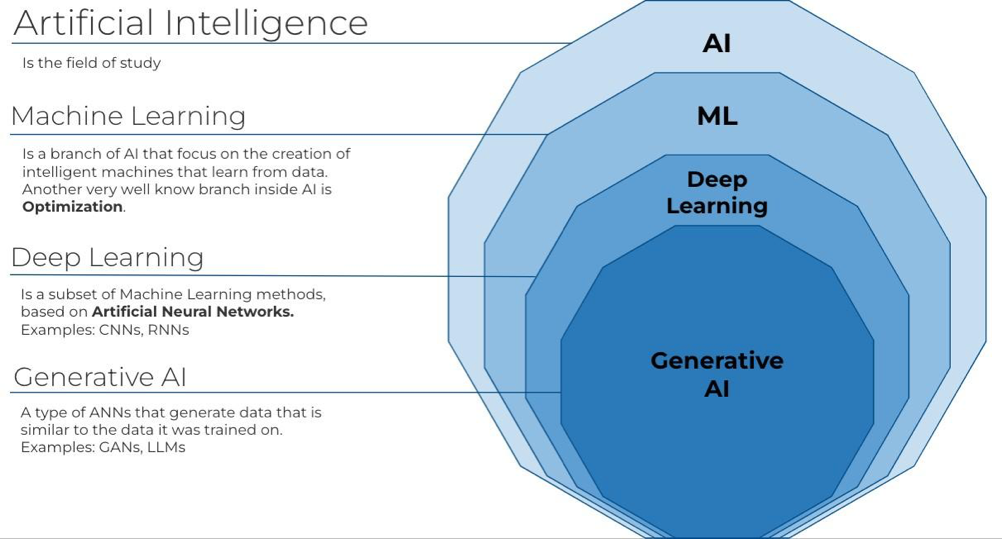
**Generative AI**

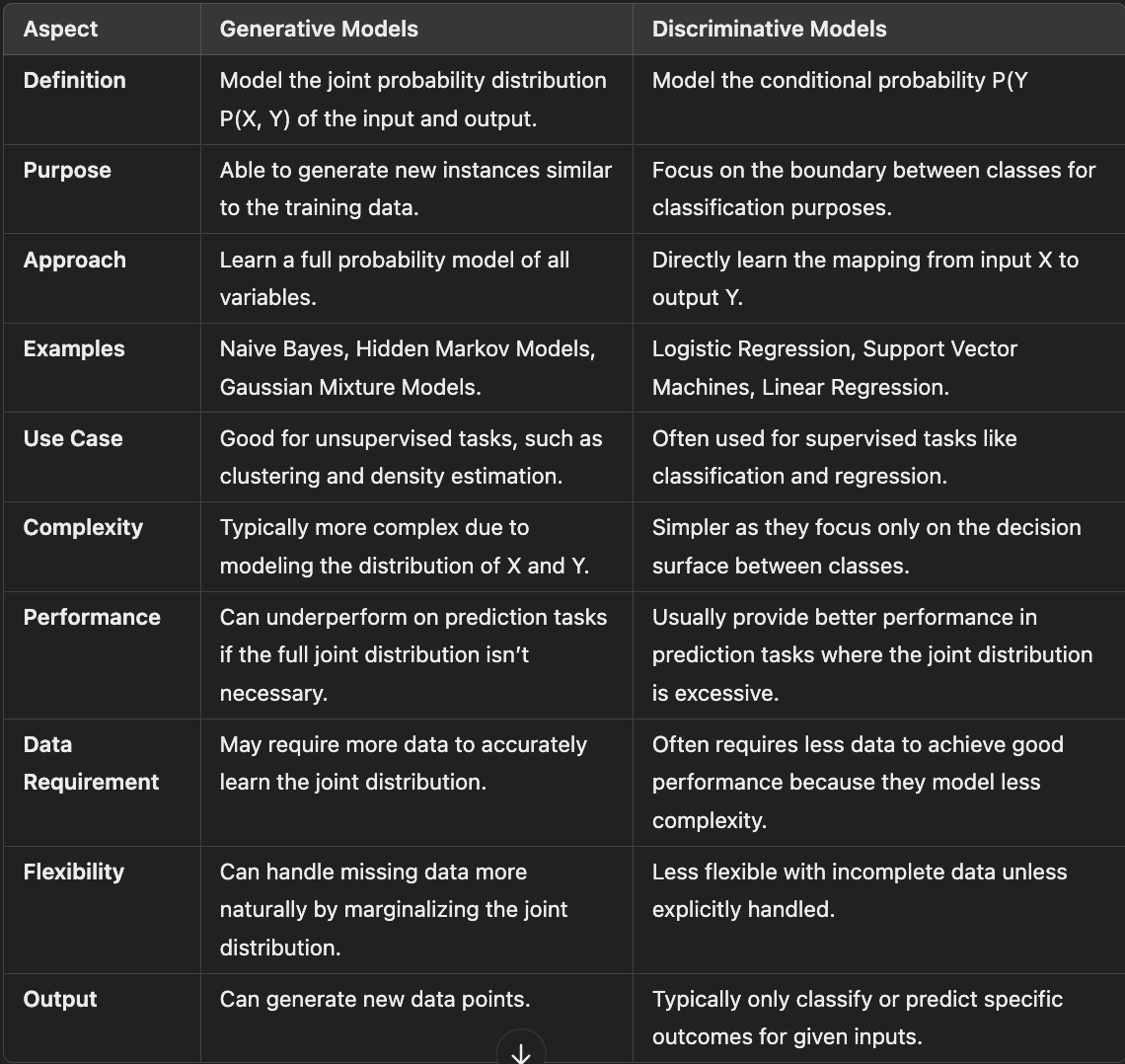
**Generative AI** generates new data based on training sample. It can generate text, audio, image, videos, data as output.

* Generative Image model
* Generative Language model

**Why generative models are required?**

* Understand complex pattern from data
* Content generation
* Build powerful application





**What data is required for training generative models?**

* We don’t need to provide labelled data. It is not possible when we have a huge amount of data.
* It tries to see the relationship between the distribution of the data.
* In Gen AI we give unstructured data to the LLM model for training purpose.

**What are LLMs?**

Foundational ML models that use DL algorithms to process and understand natural language.

Large Language Models (LLMs) are a type of machine learning model designed to process, understand, and generate human language based on vast amounts of text data. Here are some key points about LLMs:

* **Scale**: LLMs are called "large" due to the massive size of their neural network architectures, which consist of hundreds of millions to billions of parameters. These models leverage deep learning techniques and substantial compute resources.
* **Training Data**: They are trained on diverse datasets comprising a wide range of internet text. This enables them to learn a variety of language patterns, styles, and information.
* **Capabilities**: LLMs can understand and generate text, answer questions, summarize information, translate languages, and even perform specific tasks like coding.
* **Applications**: They are used in applications ranging from chatbots and virtual assistants to tools for writing assistance, content generation, and more sophisticated tasks like legal analysis and medical research.
* **Contextual Understanding**: Unlike smaller models, LLMs are capable of understanding and generating responses based on context, which allows them to maintain more coherent and contextually relevant dialogues over longer interactions.
* **Fine-tuning**: LLMs can be fine-tuned for specific tasks or industries by training them further on specialized datasets, enhancing their accuracy and relevance in particular applications.

**Examples of Large Language Models:**

* **GPT-3 and GPT-4 (OpenAI)**: Generative Pre-trained Transformer models known for their ability to generate human-like text based on prompts. GPT-4, the more advanced model, has improved abilities in reasoning and a broader range of knowledge.
* **BERT (Google)**: Bidirectional Encoder Representations from Transformers is designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers, which makes it highly effective for tasks like question answering.
* **T5 (Google)**: Text-To-Text Transfer Transformer converts all language problems into a text-to-text format, aiming to handle any task that involves generating text, including translation, summarization, and more.
* **LaMDA (Google)**: Language Model for Dialogue Applications specializes in conversational applications, providing more open-ended and nuanced responses in dialogues.
* **Llama (Meta)**: A family of models trained on a mixture of publicly available data and data licensed from third parties, designed for efficiency and scalability in various applications.

**Gen AI Pipeline**

**1. Problem Definition**

* **Purpose**: Define the problem that the generative model needs to solve.
* **Details**: This involves identifying the specific requirements, objectives, and constraints of the application. For instance, whether the model should generate text, images, music, etc.

**2. Data Collection**

* **Purpose**: Gather the data necessary to train the model.
* **Details**: This may include scraping the web, using APIs, or accessing proprietary datasets. The data must be diverse and large enough to train robust models.

**3. Data Cleaning and Pre-processing**

* **Purpose**: Prepare the data for effective training.
* **Details**: This step involves handling missing values, removing duplicates, and possibly transforming data into a format suitable for model training (e.g., tokenization for text data).

**4. Model Selection**

* **Purpose**: Choose the right generative model based on the problem.
* **Details**: This could include selecting between models like GANs, VAEs, or Transformer-based models depending on the complexity of the task and the type of data.

**5. Model Training**

* **Purpose**: Train the model to learn data distributions.
* **Details**: This involves using large datasets to train the model until it can generate new data points that are indistinguishable from real data. Training must also consider overfitting, underfitting, and computational efficiency.

**6. Model Evaluation**

* **Purpose**: Assess the performance of the model.
* **Details**: For generative models, traditional metrics (like accuracy) are not always applicable. Instead, metrics like Frechet Inception Distance (FID) for images or BLEU for text may be used to evaluate quality and diversity.

**7. Fine-tuning**

* **Purpose**: Improve the model based on feedback and evaluation.
* **Details**: Adjust model parameters, train further with more data, or tweak the architecture based on the performance metrics and user feedback.

**8. Integration**

* **Purpose**: Integrate the model into the application environment.
* **Details**: This involves embedding the model into the existing software or creating an interface for the model to interact with other components of the application.

**9. Deployment**

* **Purpose**: Deploy the model for users or stakeholders.
* **Details**: This can involve setting up a cloud infrastructure for the model to operate, ensuring scalability and security, and enabling access to the model through APIs or web interfaces.

**10. Monitoring and Maintenance**

* **Purpose**: Ensure the model performs well over time.
* **Details**: Monitor the model for drifts in data or performance decay, and perform periodic updates or retraining to maintain efficacy and relevancy.

**11. Feedback Loop**

* **Purpose**: Continuously improve the model based on user interactions.
* **Details**: Collect and analyze user interactions and feedback to refine the model, adapt to new data, or extend functionality, creating a cycle of continuous improvement.

**STEPS:**

* 1. Data Acquisition
* Available data : CSV, Audio, texts, PDFs, Docs, Excel, Image
* Other data: DB, Internet, API, Web scraping
* No data: Create your own data
* LLM models: Generate data
* If you have less data then you can perform data augmentation
  1. Data pre-processing
* Clean up
* Basic pre-processing
* Advance pre-processing
  1. Feature Engineering
* Text vectorization
  1. Modelling
* Choose model
  1. Evaluation
  2. Deployment