IT資格道場

GCP GenAI Leader 学習用テキスト

予備知識ゼロから合格ラインへ —— 読んで理解する1冊目

無料サンプル(第1章まで)

Google Cloud 公式 Generative AI Leader Exam Guide(2026-09-07 取得)準拠
2026年7月版

本書のご利用にあたって

- 本書は「IT資格道場」(www.shikaku-doujou.com)の購入者限定テキストのうち、学習用テキストの第1章までを収録した**無料サンプル**です。どなたでもご覧いただけます。
- 無料サンプルであっても著作物です。本書の転載・改変しての再配布・二次販売を禁じます。
- 本書を AI モデルの学習データとして利用すること(学習目的の入力・データセット化を含む)を禁じます。
- 製品版(学習用テキスト・頻出度別ノート・暗記用ノートの3点)は、ご購入者を識別できる透かし入りのPDFとして提供されます。
- 本PDFはスマートフォン・タブレットでの閲覧に合わせたA5版面・文字サイズで組版しています。

不正な配布を見つけた場合は info@shikaku-doujou.com までご連絡ください。

目次

この教材群の使い方(3点セット)

はじめに —— この教材の使い方

第1章 生成AI(gen AI)の基礎(ドメイン1・採点対象の 30%)

1-1 生成AIの中核的な概念とユースケース [1.1]

1-2 生成AIにおけるデータ種別とビジネス上の意味合い [1.2]

1-3 生成AIランドスケープの中核レイヤーとビジネス上の意味合い [1.3]

1-4 Google の foundation models のユースケースと強み [1.4]

1-5 補講: 取り違えやすい概念の整理と、データ種別の実務的な使われ方 [1.1/1.2/1.3/1.4]

サンプルはここまでです

この教材群の使い方(3点セット)

種類	役割
① 学習用テキスト(本書)	これだけで問題が解けるようになる本編
② 頻出度別ノート	テキストを読んだら次はこれ。頻出順に絞った問題と解説で「解ける」に橋渡し。試験直前の最終チェックにも
③ 暗記用ノート	覚えるべき要点だけを一問一答に凝縮。空き時間の反復と用語の再確認用

使い方: ① 学習用を読む → ② 頻出度別で解き方を掴む → ③ 問題演習で腕試し → ④ 頻出度別に戻って再確認(試験直前もここ)。暗記用は空き時間や用語の再確認に。

このテキストの位置づけ: 本書は①の学習用テキストです。まずはここを通読し、これだけで問題が解けるようになることを目標にしてください。

はじめに ── この教材の使い方

このテキストは、Google Cloud が実施する **Generative AI Leader 認定** の合格を目的に作られています。

この試験が問うのは、コードの書き方ではありません。問われるのは、**生成 AI の基礎概念を正しく理解し、Google Cloud の生成 AI 製品（Gemini・Gemini Enterprise・Agent Platform など）を事業課題に当てはめ、責任ある AI の観点で判断できること**です。公式の Exam guide は、想定される受験者を「業務で生成 AI を活用しようとするビジネスリーダー・実務者」としています。

本書は Google Cloud の公式教材ではなく、IT資格道場が独自に書き下ろしたオリジナルの教材です。実際に出題された問題を再現したものではありません。

試験の基本情報

項目	内容
試験名	Google Cloud Certified – Generative AI Leader
実施	Google Cloud（Kryterion のテストセンター、またはオンライン監督試験）
問題数	**50〜60問**（公式の案内）
試験時間	**90分**
出題形式	**択一（multiple choice）**
合格ライン	非公表
受験言語	英語・日本語（公式の言語一覧に日本語あり）

受験料・試験時間・出題数・試験ガイドは改定されることがあります。お申し込みの前に、Google Cloud 公式サイトで最新の情報をご確認ください。

想定される受験者

公式ガイドが想定するのは、**生成 AI を業務に取り入れる立場の人**（技術者でなくてもよい）です。本書は、**生成 AI の用語と Google Cloud の製品対応をゼロから順に積み上げる構成**にしてあります。

出題範囲の全体像 —— 4つのドメインと本書の章

ドメイン	分野	採点対象に占める割合（公式）	本書
1	生成AI(gen AI)の基礎	**30%**	第1章
2	Google Cloud の生成AI製品群	**35%**	第2章
3	生成AIモデルの出力を改善する技法	**20%**	第3章
4	生成AIソリューションを成功させるビジネス戦略	**15%**	第4章

本書の章の並びは、Exam guide の並びとまったく同じです。節見出しの末尾にある `[1.1]` のような番号は、Exam guide のセクション番号です。Google Cloud はセクションごとの割合だけを公表しており、細目単位の出題数は公表していません。本書もそれ以上の数字は書きません。

サービス名・機能名は英語のまま覚える

本文では、**製品名・機能名・用語を英語表記のまま**書いています（Gemini、Gemini Enterprise、Agent Platform、grounding、RAG、fine-tuning、Responsible AI …）。本番の設問文と選択肢に出るのはこの表記であり、日本語訳だけを覚えても画面では出会いません。英語の名前を見た瞬間に「どのドメインの、どの場面の機能か」が反射で出る状態を目指してください。

全設問に効く判断軸 ——「3つの問い」

問い	何を切り分けるか	分かれ方
問い1: どの層の話か	基礎概念／製品・サービス／事業適用／責任ある AI	同じ「精度を上げたい」でも、プロンプト・grounding・fine-tuning は層が違い、要件がどの層を指しているかで答えが決まります
問い2: 要件は何を最優先しているか	正確さ・最新性・コスト・安全性・導入の速さ	grounding／RAG／fine-tuning／プロンプト設計の選び分けは、この軸だけで一意に決まります
問い3: 責任ある AI の観点で守れているか	安全性・公平性・プライバシー・説明可能性・ガバナンス	要件を満たす案が複数あるとき、正解はより安全で、より説明できるものです

4ステップ学習法

購入者限定テキストは3冊で1セットです。

- **STEP 1（理解する）**: 本書「学習用テキスト」を通読し、4つのドメインの地図と3つの問いを頭に入れる
- **STEP 2（解き方を掴む）**: 「頻出度別ノート」で頻出度の高い論点を解いて、解き方を掴む
- **STEP 3（腕試し）**: 本サイトの問題演習で、本番と同じ択一形式に慣れる
- **STEP 4（再確認）**: 「頻出度別ノート」に戻って総ざらい。試験直前もここへ戻る

「暗記用テキスト」は本線には入れません。通勤時間や休憩時間など、**空き時間の反復と用語の再確認**に並走させてください。

それでは、第1章から始めます。

第1章 生成AI(gen AI)の基礎(ドメイン1・採点対象の 30%)

1-1 生成AIの中核的な概念とユースケース [1.1]

1-1-1 artificial intelligence・machine learning・natural language processing・generative AI の関係 [1.1]

この試験は Leader 級です。コードを書けるかではなく、用語を正しい入れ子で理解し、業務の場面で正しい打ち手を選べるかが問われます。ですからまず、いちばん外側の言葉から順に、包含関係を固めます。ここを曖昧にしたまま先へ進むと、選択肢の切り分けが最後まで安定しません。

用語の入れ子は次のとおりです。artificial intelligence(AI、人工知能)がいちばん外側の広い概念で、その内側に machine learning(ML、機械学習)があり、さらにその内側に deep learning(深層学習)があり、そのまた内側に generative AI(生成AI、gen AI)が位置します。natural language processing(NLP、自然言語処理)は「人間の言語を扱う」という応用領域の名前で、この入れ子の階層とは別軸で重なります。

用語	一言でいうと	位置づけ
artificial intelligence	人間の知的なふるまいを機械に行わせる技術の総称	いちばん広い外側の傘
machine learning	データから規則を学び取って予測・判断を行う手法群	AI の部分集合
deep learning	多層のニューラルネットワークを使う machine learning	machine learning の部分集合

用語	一言でいうと	位置づけ
generative AI	学習した分布から新しいコンテンツを生成する AI	deep learning の応用領域
natural language processing	人間の言語を機械に理解・生成させる領域	別軸の応用領域。生成AI と重なる

ここで押さえるべき対比が一つあります。従来型の AI や machine learning の多くは predictive AI(予測AI)と呼ばれ、「与えられた入力に対してラベルや数値を当てる」ことを仕事にします。これに対して generative AI は「新しい成果物そのものを作り出す」ことを仕事にします。前者の出力は分類結果やスコアで、後者の出力は文章・画像・音声・動画・コードです。

場面で確かめます。ある小売企業のカスタマーサポート部門が、問い合わせメールを「返品」「配送」「請求」の3種類に振り分けたいとします。これは分類であり predictive AI の仕事です。同じ部門が、問い合わせメールへの返信文の下書きを自動で作りたいとします。こちらは新しい文章を作るので generative AI の仕事です。

誤答肢の切り方も見ておきます。「generative AI は machine learning とは別系統の技術である」という選択肢は誤りです。generative AI は machine learning の内側にあるからです。「natural language processing は generative AI の一種である」という選択肢も誤りです。NLP は生成AI 登場よりずっと前から存在する領域で、包含関係が逆になっています。「deep learning を使わない generative AI は存在しない」という断定も、試験の文脈では避けるべき言い切りです。現在の主要な生成AI はほぼ deep learning ベースですが、問われているのは階層の理解であって排他の証明ではありません。

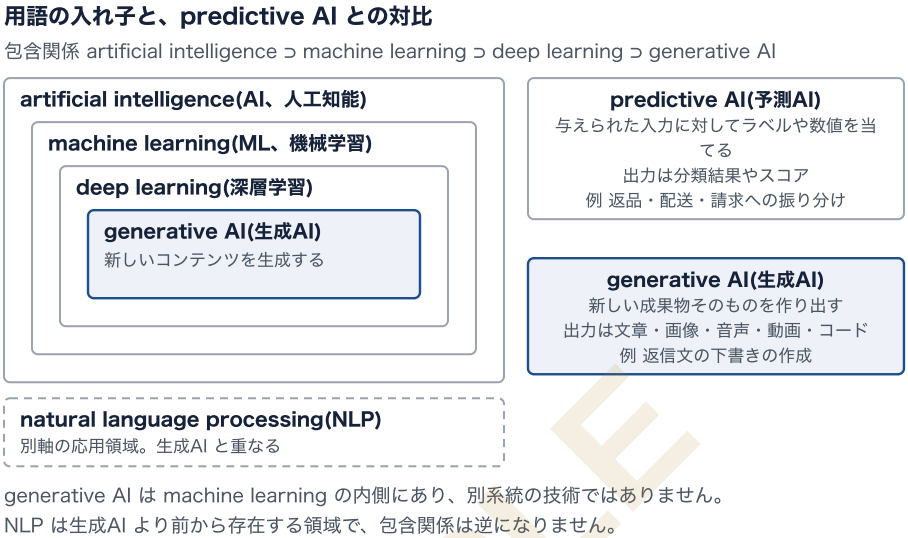

図 1-1　用語の入れ子と、predictive AI との対比

1-1-2 foundation models と large language models [1.1]

foundation model(基盤モデル)は、この試験の中心にある用語です。定義は「膨大で多様なデータで大規模に事前学習され、幅広い下流タスクに適応させられる汎用モデル」です。ポイントは「特定の一つの仕事のために作られていない」ことにあります。一度作った巨大なモデルを、要約にも翻訳にも分類にも下書き生成にも転用できる、という点が従来の「タスクごとに専用モデルを作る」やり方との決定的な違いです。

large language models(LLM、大規模言語モデル)は、foundation model のうち、言語を主たる対象とするものを指します。つまり foundation model が上位概念で、LLM はその一種です。試験では「foundation model = LLM」と読める選択肢が誤答肢として置かれやすいので、上位と下位を取り違えないでください。

foundation model が広く使えるのは、事前学習(pre-training)と適応(adaptation)を分けているからです。事前学習では、インターネット規模の大量のテキストや画像を使い、「次に来るものを予測する」といった自己教師的な目標で長時間・高コストの学習を行います。適応の段階では、prompt engineering や fine-tuning といった比較的軽い手段で、自社の業務に寄せます。この二段構えが、企業が自前で巨大モデルを一から学習しなくてよい理由です。

用語	一言定義	よくある取り違え
foundation model	大規模事前学習された汎用モデル。多数の用途に適応できる	LLM と同義だと思ってしまう
large language models	言語を主対象とする foundation model	画像生成モデルまで LLM と呼んでしまう
multimodal foundation models	複数のモダリティを同時に扱える foundation model	テキスト専用モデルに画像を渡せると考えてしまう
diffusion models	ノイズから段階的に復元して画像・動画を生成するモデル	LLM と同じ仕組みだと考えてしまう

multimodal foundation models(マルチモーダル基盤モデル)は、テキストだけでなく画像・音声・動画・PDF といった複数の modality(モダリティ、入出力の種類)を同時に扱えるモデルです。Google の Gemini はこの multimodal foundation models の代表例で、画像とテキストを同じ入力の中に混ぜて渡し、両方を踏まえた回答を得られます。

multimodal がビジネスで効く場面を挙げます。保険会社の損害査定部門が、事故車両の写真と保険約款の文章を同時にモデルへ渡し、「この損傷が約款のどの条項に当たるか」を下書きさせる。製造業の品質保証部門が、検査画像

と作業手順書を同時に渡し、逸脱箇所を説明させる。どちらも、画像だけ・文章だけでは成立しません。

diffusion models(拡散モデル)は、画像や動画の生成で使われるモデルの種類です。仕組みは、学習時に画像へ少しずつノイズを加えていき、生成時にはその逆をたどってノイズから画像を復元する、というものです。LLM が「次のトークンを予測する」のに対し、diffusion models は「ノイズを除去していく」ので、生成の原理そのものが異なります。Imagen や Veo のような画像・動画の生成は、この系譜の技術に支えられています。

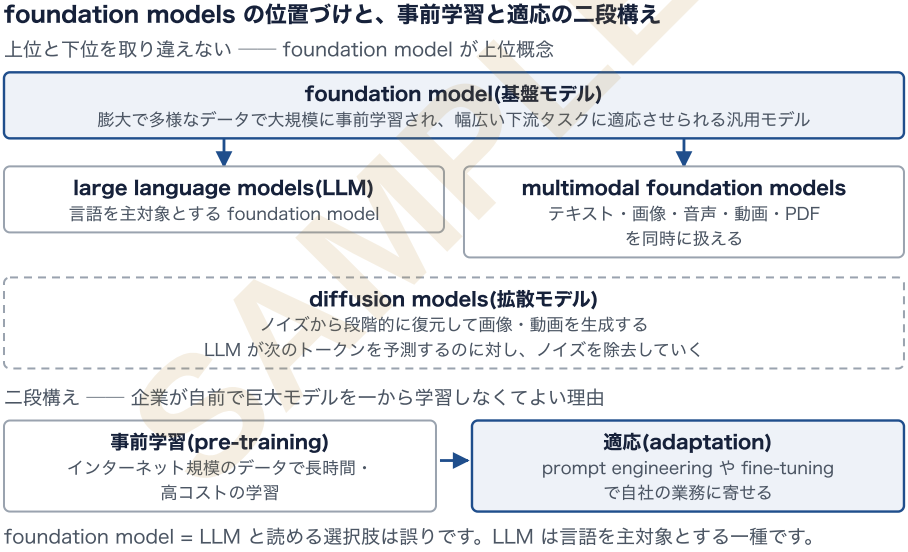

図 1-2　foundation models の位置づけと、事前学習と適応の二段構え

1-1-3 tokens・context window・parameters [1.1]

LLM はテキストをそのまま文字として扱っているのではありません。tokens(トークン)という単位に切り分けて扱います。token は単語よりも細かいことが多く、英語ではおおむね単語の一部、日本語では1〜数文字が1 token になり

ます。料金も速度も制限も、この token を単位に決まるので、Leader として費用を見積もるなら token が単位だと覚えておく必要があります。

context window(コンテキストウィンドウ)は、モデルが一度の推論で参照できる token の最大量です。入力(プロンプトや添付文書)と出力を合わせた上限として扱われます。context window が大きいほど、長い契約書や大量の議事録をそのまま渡して扱えます。逆に小さければ、文書を分割したり、RAG のように必要部分だけ取り出して渡す設計が必要になります。

context window の大きさは、業務設計に直結します。ある法務部門が300ページの契約書を丸ごと読ませて要点を抽出したいなら、大きな context window を持つモデルを選ぶのが素直です。一方、社内の数十万件のナレッジ記事から答えたいのであれば、いくら context window が大きくても全件は入りません。この場合は grounding や retrieval-augmented generation(RAG)で必要な数件だけを取り出して渡します。「context window を大きくすれば RAG は不要になる」という選択肢は誤答肢の定番です。

parameters(パラメータ)は、モデルが学習によって獲得した重みの数で、モデルの規模を表す指標として使われます。一般に parameters が多いほど能力が高くなりやすい一方、推論のコストと遅延も増えます。ですから Leader の判断としては「常に最大のモデルを選ぶ」ではなく、「求める品質を満たす最小のモデルを選ぶ」が基本になります。

1-1-4 prompt engineering と prompt tuning の違い [1.1]

この二つは名前が似ていますが、まったく別の操作です。試験では並べて出されるので、ここで確実に分けます。

prompt engineering(プロンプトエンジニアリング)は、モデルの重みを一切変更せず、入力する指示文の書き方を工夫して出力を改善する営みです。役割を与える、出力形式を指定する、例を添える、手順を分けて考えさせる。どれも入力側の工夫であり、モデル自体には手を加えません。追加の学習コストがかからず、すぐ試せて、すぐ戻せるのが利点です。

prompt tuning(プロンプトチューニング)は、parameter-efficient な調整手法の一つで、モデル本体の重みは凍結したまま、入力に付加する少量の学習可能なベクトル(ソフトプロンプト)だけを学習させます。人間が読める文章を書くのではなく、機械が学習した内部表現を前置きする点が prompt engineering との違いです。少量のデータと計算で、特定タスクへの適応が得られます。

観点	prompt engineering	prompt tuning	fine-tuning
モデル重みの変更	しない	しない(付加ベクトルのみ学習)	する(全部または一部)
必要なデータ	例が数件あればよい	数百〜数千件程度	まとまった学習データ
コストと時間	最小。即時	中程度	最大
取り消しやすさ	容易	容易	版管理が必要
主な狙い	指示の明確化・出力形式の制御	特定タスクへの軽量適応	語調・専門性・形式の恒常的な作り込み

Leader としての順序は「まず prompt engineering、次に grounding や RAG、それでも足りなければ prompt tuning や fine-tuning」です。コストが

低く可逆な手段から試すのが Google Cloud の推奨する現実的な進め方であり、この順序は第3章でも繰り返し出てきます。

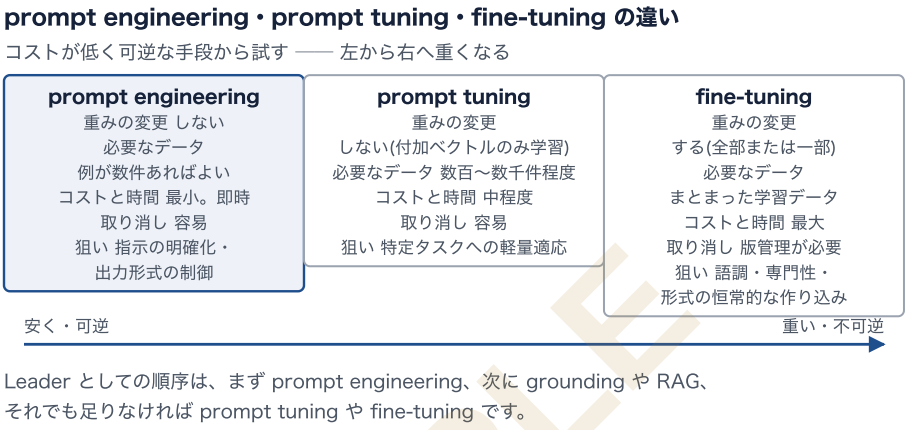

図 1-3　prompt engineering・prompt tuning・fine-tuning の違い

1-1-5 machine learning approaches: supervised・unsupervised・reinforcement [1.1]

台帳が求める machine learning approaches(機械学習の学習方式)は三つです。supervised learning(教師あり学習)、unsupervised learning(教師なし学習)、reinforcement learning(強化学習)。それぞれ「何を手がかりに学ぶか」が違います。

supervised learning は、入力と正解ラベルの組を大量に与えて、入力から正解を当てる関数を学ばせる方式です。labeled data(ラベル付きデータ)が必要になります。用途は classification(分類)と regression(回帰)に大別されます。分類の例は「この問い合わせは返品か配送か請求か」、回帰の例は「この物件の販売価格は何円か」です。

unsupervised learning は、正解ラベルのない unlabeled data(ラベルなしデータ)から、データ自身が持つ構造を見つけ出す方式です。代表的な用途は clustering(クラスタリング、似たもの同士のグループ分け)、dimensionality reduction(次元削減)、anomaly detection(異常検知)です。「顧客をあらかじめ決めた区分に当てはめる」のではなく「顧客が自然にいくつのまとまりに分かれるかを見つける」のが unsupervised です。

reinforcement learning は、環境の中で行動し、得られる報酬(reward)を最大化するように試行錯誤で方策を学ぶ方式です。正解の一手を教えるのではなく、結果の良し悪しだけを返します。ロボット制御、在庫や価格の最適化、ゲームのプレイなどが典型です。生成AI の文脈では、reinforcement learning from human feedback(RLHF)として、人間の好ましさ評価を報酬に使い、モデルの応答を人間の期待に沿わせる目的で使われます。

方式	必要なデータ	学習の手がかり	代表的な用途
supervised	labeled data	正解ラベル	classification、regression、需要予測、スパム判定
unsupervised	unlabeled data	データ自体の構造	clustering、次元削減、anomaly detection、顧客セグメント発見
reinforcement	環境と報酬設計	行動に対する報酬	ロボット制御、価格最適化、RLHF による応答の調整

誤答肢の切り方です。「顧客を似た者同士のグループに分けたい。ただし事前にグループの定義は決めていない」という場面で supervised を選ぶのは誤りです。事前定義がないので正解ラベルが作れず、これは clustering、すなわ

ち unsupervised だからです。逆に「過去の申込データに承認・却下の結果が全部付いている。新規申込の可否を予測したい」で unsupervised を選ぶのも誤りです。ラベルがすでに揃っているので supervised が素直です。

semi-supervised learning(半教師あり学習)という中間の方式もあります。少量の labeled data と大量の unlabeled data を組み合わせる方式で、「ラベル付けのコストが高いが、生データは大量にある」という現実的な状況で使われます。試験では主要三方式が問われますが、この中間形の存在も知っておくと選択肢の読み違いを避けられます。

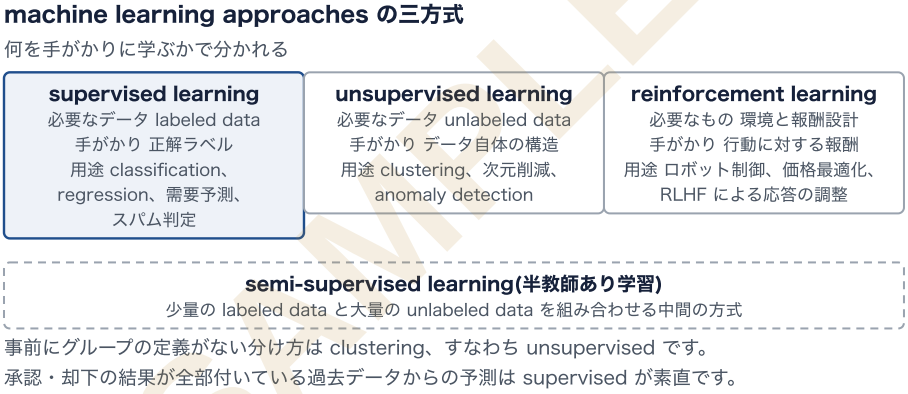

図 1-4　machine learning approaches の三方式

1-1-6 machine learning lifecycle の5段階と Google Cloud のツール [1.1]

台帳は machine learning lifecycle(機械学習ライフサイクル)の段階として、data ingestion、data preparation、model training、model deployment、model management の五つを挙げ、各段階の Google Cloud tools を特定できることを求めています。順に見ます。

data ingestion(データ取り込み)は、社内外の各所に散らばるデータを集めて一箇所に流し込む段階です。基幹システムのデータベース、SaaS のログ、センサーからのストリーム、PDF や画像といったファイル。形も速度もばらばらのものを受け止めます。Google Cloud では、オブジェクトを置く先として Cloud Storage、ストリーミングの受け口として Pub/Sub、バッチとストリームの取り込み処理として Dataflow、GUI 中心のパイプライン構築として Cloud Data Fusion、そして分析用の受け皿として BigQuery が使われます。

data preparation(データ準備)は、集めたデータを学習や推論に使える形に整える段階です。欠損の処理、重複の除去、形式の統一、ラベル付け、特徴量の作成が含まれます。ここは全工程の中でもっとも手間がかかる段階で、しかも成果物の品質を最終的に左右します。Google Cloud では BigQuery(SQL による変換)、Dataflow、Dataprep、そして特徴量を管理する Feature Store(現在は Agent Platform Feature Store として提供)が使われます。

model training(モデル学習)は、準備したデータでモデルを学習させる段階です。生成AI の文脈では、一からの事前学習を行う企業はまれで、実際には foundation model に対する fine-tuning や蒸留(distillation)が中心になります。Google Cloud では Agent Platform(旧 Vertex AI)の training 機能、AutoML(コードを書かずに学習させる)、BigQuery ML(SQL だけでモデルを作る)、そして基盤としての TPU と GPU が使われます。

model deployment(モデルデプロイ)は、学習したモデルを実際に使える形で提供する段階です。オンラインで即時に応答する online prediction(オンライン推論)と、まとめて処理する batch prediction(バッチ推論)の二つの提供形態があります。Google Cloud では Agent Platform のエンドポイント、Agent Runtime(エージェントの実行基盤)、そしてコンテナ実行基盤としての Cloud Run が使われます。

model management(モデル管理)は、動き出した後を見続ける段階です。versioning(バージョン管理)、performance tracking(性能追跡)、drift monitoring(ドリフト監視)、再学習の判断、security patches and updates(セキュリティ更新)が含まれます。Google Cloud では Agent Platform の Model Registry、Model Monitoring、Evaluation、そして横断的な監視として Cloud Monitoring と Cloud Logging が使われます。

段階	やること	主な Google Cloud tools
data ingestion	各所からデータを集める	Cloud Storage、Pub/Sub、Dataflow、Cloud Data Fusion、BigQuery
data preparation	整形・欠損処理・ラベル付け・特徴量作成	BigQuery、Dataflow、Dataprep、Feature Store
model training	学習・fine-tuning・蒸留	Agent Platform(旧 Vertex AI)、AutoML、BigQuery ML、TPU、GPU
model deployment	推論できる形で提供する	Agent Platform エンドポイント、Agent Runtime、Cloud Run
model management	監視・版管理・再学習・更新	Model Registry、Model Monitoring、Evaluation、Cloud Monitoring、Cloud Logging

生成AI の時代になっても、このライフサイクルが消えるわけではありません。むしろ、model training の比重が下がって data preparation と model management の比重が上がった、と捉えるのが実務に近い見方です。自社データの整備と、出た後の監視。この二つに投資が要る、というのが Leader として持っておくべき感覚です。

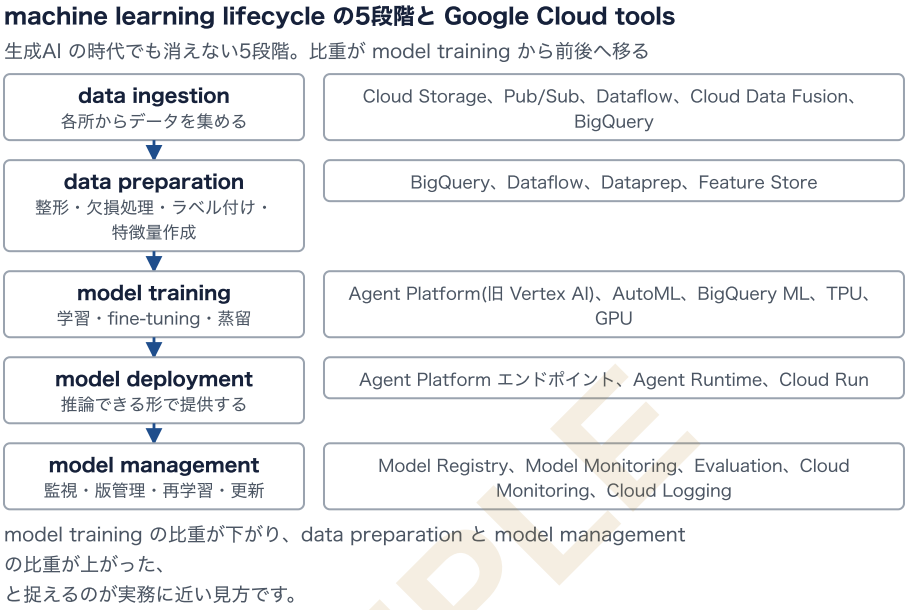

図 1-5　machine learning lifecycle の5段階と Google Cloud tools

1-1-7 業務ユースケースに合う foundation model の選び方 [1.1]

台帳は、foundation model を選ぶ際の判断軸として modality、context window、security、availability and reliability、cost、performance、fine-tuning、customization を挙げています。一つずつ、何を見るのかを確認します。

modality(モダリティ)は、扱える入出力の種類です。テキストだけでよいのか、画像も入力するのか、動画を出力するのか。求める modality を扱えないモデルは、他の条件がいくら良くても候補から外れます。ここが最初の足切りになります。

context window は前述のとおり、一度に渡せる token の量です。長大な文書をまとめて扱う業務なら大きいものが要ります。逆に短い問い合わせ応答

なら、大きな context window に費用を払う意味は薄くなります。

security(セキュリティ)は、データがどこで処理され、どこに保存され、モデルの再学習に使われるのかという論点です。企業利用では「入力データを提供元のモデル学習に使わない」ことが要件になることが多く、Google Cloud はこの点をエンタープライズ向けの前提として提供しています。データの所在地(リージョン)や暗号化、アクセス制御も同じ軸に含まれます。

availability and reliability(可用性と信頼性)は、必要なリージョンで使えるか、SLA があるか、割り当て(quota)が業務量に足りるかという論点です。試作では動いたのに本番の同時アクセスに耐えられない、という失敗はここを見ていないときに起きます。

cost(コスト)は、原則として入力 token と出力 token の量で決まります。ですから「モデルの単価」だけでなく「1リクエストあたり何 token 使うか」「1日に何リクエストあるか」を掛け合わせて見ます。プロンプトに毎回大量の文書を貼り付ける設計は、単価の安いモデルを選んでも総額が跳ね上がります。

performance(性能)は、品質・遅延・スループットの三つに分けて考えます。品質は求めるタスクでの正確さ、遅延は1件あたりの応答時間、スループットは単位時間あたりの処理量です。対話型のアシスタントは遅延が効き、夜間のバッチ処理はスループットが効きます。同じ「性能」でも、業務によって見るべき指標が違います。

fine-tuning と customization(カスタマイズ)は、そのモデルを自社向けに調整できるかという軸です。fine-tuning に対応しているか、grounding で自社データに接続できるか、safety settings を業務に合わせて調整できるか。将来の伸びしろを含む判断になります。

判断軸	具体的に見ること	外すとどうなるか
modality	扱う入出力の種類が合うか	そもそも要件を満たせない
context window	一度に渡す文書量が入るか	分割設計や RAG が別途必要になる
security	データの所在・学習利用・アクセス制御	社内規程や法規制に抵触する
availability and reliability	リージョン・SLA・quota	本番の負荷で止まる
cost	token 単価 × 使用量	想定外の請求になる
performance	品質・遅延・スループット	体験が悪化する、処理が終わらない
fine-tuning / customization	調整・接続の余地	品質改善の打ち手が尽きる

場面で確かめます。ある製造業の現場支援チームが、作業者がスマートフォンで撮った不具合写真と、手元の作業手順書のテキストを一緒に渡して、対処方法を提示させたいとします。まず modality で multimodal が必須になり、テキスト専用モデルは落ちます。現場で待たせられないので performance のうち遅延を重視し、巨大な最上位モデルより軽量で高速なモデルを優先します。手順書は社内文書なので security と grounding の可否を確認します。この三段階で候補はかなり絞れます。

誤答肢の切り方も見ます。「最も parameters の多いモデルを選ぶ」は、コストと遅延を無視しているので誤りです。「無料枠が最大のモデルを選ぶ」も、業務要件から出発していないので誤りです。Leader 級の問いでは、判断が業務要件から始まっているかどうかが正誤の分かれ目になります。

図 1-6　業務ユースケースに合う foundation model を選ぶ八つの軸

1-1-8 生成AI が create・summarize・discover・automate するビジネスユースケース [1.1]

台帳は、生成AI が create(創る)、summarize(要約する)、discover(見つける)、automate(自動化する)ことのできるビジネスユースケースの特定を求めています。そして例として text generation、image generation、code generation、video generation、data analysis、personalized user experience を挙げています。四つの動詞に沿って整理します。

create は、新しい成果物を作る用途です。text generation(テキスト生成)は、商品説明文、メール返信の下書き、求人票、社内告知の作成に使われます。image generation(画像生成)は、広告バナー、商品イメージ、プレゼン資料の挿絵に使われます。code generation(コード生成)は、関数の実装、テストコード、SQL、設定ファイルの作成に使われます。video generation(動画生成)は、短尺の広告、商品紹介、研修用の映像に使われます。

summarize は、大量の入力を短く畳む用途です。長い会議の議事録から決定事項と宿題を抜き出す。数百件のカスタマーレビューから頻出の不満を三つにまとめる。100ページの技術仕様書から経営会議向けに1ページの要約を作る。入力が人間の読める量を超えているときほど効果が出ます。

discover は、埋もれている情報や関係を見つける用途です。社内の膨大なドキュメントから該当する規程を探す。data analysis(データ分析)の文脈では、自然言語の質問から SQL を組み立てて BigQuery に問い合わせ、結果を言葉で説明させる、という使い方が典型です。「探す」ことを人間の検索語彙に依存させず、意味で当てにいけるのが従来の全文検索との違いです。

automate は、判断を含む定型業務を機械に回させる用途です。問い合わせの一次対応、請求書からの項目抽出と基幹システムへの登録、レポートの定期作成。ここは第2章で扱う AI agents の領域と地続きになります。

personalized user experience(パーソナライズされたユーザー体験)は、上の四つを横断する成果です。同じ商品ページでも顧客の閲覧履歴に合わせて説明の切り口を変える、同じ研修でも受講者の理解度に合わせて例題を変える。生成AI は「一人ひとりに合わせて作り直す」ことのコストを下げるので、従来は採算が合わなかったパーソナライズが成立します。

動詞	何をするか	部門と業務の例
create	新しい成果物を作る	マーケティング部門が text generation と image generation で広告案を量産する
summarize	長い入力を短く畳む	法務部門が契約書の変更点を要約して要注意条項を洗い出す

動詞	何をするか	部門と業務の例
discover	埋もれた情報・関係を見つける	営業部門が過去の提案書から類似案件を探し出す。data analysis で売上の異常を説明させる
automate	判断を含む定型業務を回す	サポート部門が一次回答を自動生成し、複雑な案件だけ人間へ回す

Leader としての注意点を一つ添えます。生成AI を導入する価値は「作れること」自体ではなく、「これまで人手の制約でやれなかった量と速度が出せること」にあります。ですから効果測定も、品質の主観評価だけでなく、処理件数、1件あたりの所要時間、対応可能時間帯といった量の指標で見るのが筋です。この考え方は第4章の効果測定につながります。

1-2 生成AIにおけるデータ種別とビジネス上の意味合い [1.2]

この節の topics は、台帳上 1.1 と 1.2 の両方に同じ内容で掲載されています。原文の重複をそのまま扱い、ここでまとめて詳しく述べます。

1-2-1 data quality と data accessibility の特性と重要性 [1.1/1.2]

生成AI の成果は、モデルの良し悪しよりも、渡すデータの質で決まる場面が多くあります。台帳は data quality(データ品質)と data accessibility(データアクセス性)の特性として completeness、consistency、relevance、availability、cost、format を挙げています。順に、何が欠けるとどう壊れるかを見ます。

completeness(完全性)は、必要な項目や期間が欠けなく揃っているかです。顧客マスタの3割で業種が空欄なら、業種別の分析も業種別のパーソナライズも成り立ちません。生成AI に渡した場合、モデルは欠けていることを明示せず、もっともらしく埋めてしまうことがあります。欠損は hallucination(ハルシネーション、もっともらしい誤りの生成)の温床になります。

consistency(一貫性)は、同じものが同じ表現で書かれているかです。同一顧客が「株式会社A」「(株)A」「A社」と三通りで登録されていれば、名寄せができず、検索でも取りこぼします。単位が「円」と「千円」で混在していれば、集計値は1000倍ずれます。

relevance(関連性)は、そのデータが解こうとしている課題に関係しているかです。量が多ければ良いわけではありません。10年前に廃止した製品のマニュアルを grounding の対象に含めれば、モデルはそれを根拠に現行製品の誤った手順を答えます。古い・無関係なデータを混ぜないことは、足すことと同じくらい重要です。

availability(可用性)は、必要なときに実際に取り出せるかです。データは存在するが、権限が下りない、担当者しか触れない、月次でしか更新されない。こうした状態は「データがない」のと実務上は変わりません。

cost(コスト)は、そのデータを取得・保管・加工・更新するのにかかる費用です。外部データの購入費、保管費用、ラベル付けの人件費、パイプラインの維持工数を含みます。品質を上げるほど費用は上がるので、どこまで上げるかは投資判断になります。

format(形式)は、機械が扱える形になっているかです。紙の書類をスキャンしただけの画像 PDF は、そのままでは検索も抽出もできません。Document AI API のような手段でテキスト化して初めて使えるようになります。

特性	満たされないと起きること	打ち手の例
completeness	欠損を埋めた誤答、偏った分析	必須項目の定義、入力時のバリデーション
consistency	名寄せ失敗、単位ずれ、検索の取りこぼし	表記ゆれの正規化、マスタ統合
relevance	古い情報を根拠にした誤答	対象期間・対象製品の絞り込み、廃止文書の除外
availability	使いたいときに使えない	アクセス権の設計、更新頻度の見直し
cost	投資対効果が合わない	対象範囲の限定、優先度の高いデータから着手
format	機械が読めない	Document AI API 等によるテキスト化、形式統一

data accessibility(データアクセス性)は、availability と重なりますが、より広く「必要な人と必要なシステムが、必要な範囲だけを、適切な権限のもとで取り出せる状態」を指します。ここで重要なのは、アクセスを広げることと守ることが両立していなければならない点です。全社員が全データを見られる状態はアクセス性が高いようで、実際には情報漏えいのリスクを負っており、企業では採用できません。Gemini Enterprise の検索が permissions-aware(権限を踏まえる)である、すなわち利用者本人が本来アクセスできる範囲の結果しか返さない、という設計は、この両立を製品側で担保する仕組みです。

1-2-2 structured data と unstructured data の違いと実例 [1.1/1.2]

structured data(構造化データ)は、あらかじめ決められたスキーマ、すなわち行と列と型が定義された形で格納されているデータです。リレーショナルデー

タベースのテーブル、CSV、スプレッドシート、BigQuery のテーブルが該当します。列の意味が定義されているので、SQL で集計・結合・絞り込みができます。

unstructured data(非構造化データ)は、決まったスキーマを持たないデータです。文書、メール、契約書 PDF、議事録、画像、音声、動画、ソースコード、チャットログが該当します。企業が保有するデータの大部分はこちらだと言われており、従来はほとんど活用されずに眠っていました。

semi-structured data(半構造化データ)という中間もあります。JSON、XML、ログファイルのように、厳密なスキーマはないがタグやキーで構造の手がかりがあるものです。

種別	定義	実例	従来の扱いやすさ
structured data	行・列・型が定義済み	販売実績テーブル、顧客マスタ、在庫数、センサー値のテーブル	高い。SQL で集計できる
semi-structured data	部分的に構造がある	JSON の API 応答、アプリのログ、XML の設定	中程度
unstructured data	スキーマなし	契約書 PDF、問い合わせメール、会議録音、製品写真、監視カメラ映像	低い。人が読むしかなかった

生成AI がビジネスにもたらしたいちばん大きな変化は、この unstructured data を扱えるようになったことです。従来、社内の膨大な文書・音声・画像は「保管はしているが分析できない」資産でした。生成AI と embeddings、vector search を組み合わせれば、意味で検索し、要約し、構造化データへ変

換できます。眠っていた資産が動き出す、という点が Leader として説明すべき価値です。

実務での組み合わせも押さえておきます。問い合わせメール(unstructured)から「製品名・不具合種別・緊急度」を抽出してテーブル(structured)にすれば、以降は SQL で傾向分析ができます。unstructured から structured への変換が、生成AI の代表的な実務パターンの一つです。

1-2-3 labeled data と unlabeled data の違い [1.1/1.2]

labeled data(ラベル付きデータ)は、各データ点に「正解」が付与されているデータです。画像に「猫」「犬」というタグが付いている、メールに「スパム」「通常」の判定が付いている、申込に「承認」「却下」の結果が付いている。この正解があるからこそ supervised learning が成立します。

unlabeled data(ラベルなしデータ)は、正解が付いていない生のデータです。撮っただけの画像、記録しただけのログ、蓄積しただけの文書。企業が持つデータの大半はこちらです。unsupervised learning はこちらを対象にします。

観点	labeled data	unlabeled data
正解の有無	ある	ない
対応する学習方式	supervised learning	unsupervised learning
入手のしやすさ	難しい。人手のラベル付けが要る	容易。すでに大量にある
コスト	高い。専門知識が要る領域ではさらに高い	低い
例	承認・却下が記録された過去の与信申込	まだ判定していない申込の記録

構造化・非構造化との関係を混同しないでください。この二つは別の軸です。structured data であってもラベルが付いていないことはあり(売上テーブルに「異常か否か」の列はない)、unstructured data であってもラベルが付いていることはあります(過去の問い合わせメールに担当者が分類を付けている)。「構造化データ = ラベル付きデータ」と読める選択肢は誤りです。

ビジネス上の意味合いを一つ加えます。ラベル付けは高価で時間がかかるため、これが生成AI 導入の隠れたボトルネックになりがちです。だからこそ、foundation model を使う方式は魅力的です。foundation model は膨大な unlabeled data で事前学習されており、prompt engineering や few-shot の例示だけで多くのタスクをこなせるので、大量のラベル付けなしに始められます。「自社にラベル付きデータがないから生成AI は使えない」という思い込みは、Leader として正しておくべき誤解です。

1-3 生成AIランドスケープの中核レイヤーとビジネス上の意味合い [1.3]

生成AI のエコシステムは、下から上へ五つのレイヤーに積み上がっています。台帳が挙げる Infrastructure、Models、Platforms、Agents、Applications です。このレイヤー分けは、「自社はどの層で戦い、どの層は買うのか」を決めるための地図として使います。

1-3-1 Infrastructure(インフラ層) [1.3]

いちばん下は、モデルを学習・推論させる計算基盤です。データセンター、電力と冷却、ネットワーク、そして AI 専用のアクセラレータが含まれます。Google Cloud では、自社設計の TPU(Tensor Processing Unit)、GPU、それらを大規模に束ねる AI Hypercomputer(AI ハイパーコンピュータ)というアーキテク

チャ、そして cloud computing(クラウドコンピューティング)としての提供形態が該当します。

ビジネス上の意味合いは明確です。この層を自前で持つには巨額の設備投資と、電力・冷却・調達の長期計画が要ります。ほとんどの企業にとって、ここは所有ではなく利用する層です。cloud computing の本質的な価値、すなわち初期投資なしに使った分だけ払い、必要なときに必要な量へ伸縮できるという性質が、いちばん効くのがこの層です。

1-3-2 Models(モデル層) [1.3]

その上に、学習済みの foundation model が乗ります。Google の Gemini、Gemma、Imagen、Veo、そしてサードパーティやオープンソースのモデル群です。この層の企業は、モデルを学習させて提供することを事業にしています。

ビジネス上の意味合いは、「モデルを自分で作るか、既にあるものを使うか」という判断です。foundation model の事前学習には、巨大な計算資源と大規模なデータと専門人材が要ります。ですから大多数の企業にとって、ここも作る層ではなく選ぶ層です。選ぶときの軸は 1-1-7 で見た八つになります。

1-3-3 Platforms(プラットフォーム層) [1.3]

モデルを業務で使える形にまとめる層です。モデルへの API アクセス、fine-tuning、grounding、評価、デプロイ、監視、セキュリティ、課金がここに集約されます。Google Cloud では Agent Platform(旧 Vertex AI)がこの層に当たり、Model Garden によるモデルの品揃え、Agent Studio による開発、Agent Runtime による実行、Evaluation による評価がまとまっています。

ビジネス上の意味合いは、開発の速度とガバナンスの両立です。プラットフォームを使わずに個々のモデル API を直接叩く構成でも動くものは作れますが、権限管理、ログ、コスト配賦、モデルの入れ替えといった運用の仕事が各チ

ームに散らばります。組織として複数のユースケースを回すなら、この層を統一するのが定石です。

1-3-4 Agents(エージェント層) [1.3]

モデルに目標を与え、自律的に手順を組み立て、tools(ツール)を使って外部と関わりながらタスクを完了させる仕組みの層です。単発の質問応答ではなく、複数のステップと外部システムの呼び出しを含む点が特徴です。この層は第2章の 2-5 で詳しく扱います。

ビジネス上の意味合いは、自動化の範囲が「文章を作る」から「業務を進める」へ広がることです。同時に、外部システムへ書き込む権限を持つ以上、権限設計と監査、そして human in the loop(HITL、人間の確認を挟むこと)の設計が不可欠になります。

1-3-5 Applications(アプリケーション層) [1.3]

いちばん上は、業務利用者が直接触れるアプリケーションです。Gemini アプリ、Gemini for Google Workspace、Gemini Code Assist、そして企業が自社の顧客・社員向けに作るアプリがここに入ります。

ビジネス上の意味合いは、価値が実際に生まれるのはこの層だけだ、ということです。下の四層にいくら投資しても、利用者が日常の仕事の中で使わなければ成果は出ません。ですから導入の順序としては、まず既製のアプリケーション層(Gemini for Google Workspace など)で価値を体験させ、それから独自の開発へ進むのが、立ち上がりの速い進め方になります。

レイヤー	何があるか	Google Cloud の例	自社の関わり方
Applications	利用者が触る画面・体験	Gemini アプリ、Gemini for Google Workspace、Gemini Code Assist	買う、または作る。価値が生まれる場所
Agents	目標を受け取り自律的に実行	Agent Platform で作る custom agents、Agent Gallery の prebuilt agents	作ることが多い。権限設計が要点
Platforms	モデルの提供・調整・運用・統制	Agent Platform(旧 Vertex AI)、Model Garden、Agent Studio	使う。統一するとガバナンスが効く
Models	学習済みの foundation model	Gemini、Gemma、Imagen、Veo、サードパーティ、オープンモデル	選ぶ。作るのは例外的
Infrastructure	計算・保管・ネットワーク	TPU、GPU、AI Hypercomputer、データセンター	利用する。所有はしない

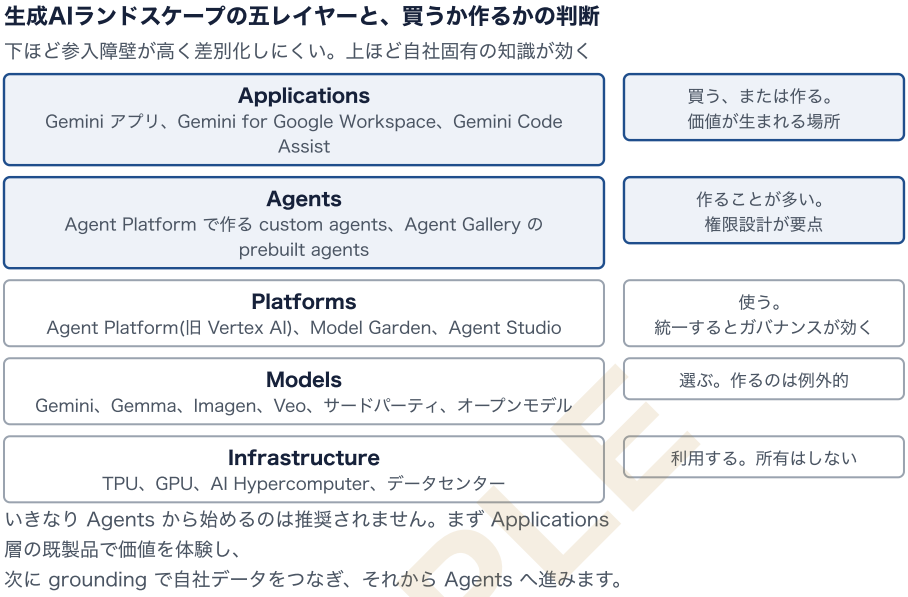

図 1-7　生成AIランドスケープの五レイヤーと、買うか作るかの判断

1-4 Google の foundation models のユースケースと強み [1.4]

台帳は Gemini、Gemma、Imagen、Veo の四つを挙げています。この四つを混同しない、というのが本節の到達点です。

1-4-1 Gemini [1.4]

Gemini は Google のフラッグシップとなる multimodal foundation models のファミリーです。テキストに加えて、画像、音声、動画、PDF などを入力として同時に扱い、テキストを生成します。汎用の推論、要約、コード生成、対話、文書からの抽出まで、幅広く使えます。

Gemini はファミリーで提供され、用途に応じて選び分けます。高度な推論や複雑なタスク向けの Pro 系、速度とコスト効率を重視した Flash 系、さらに軽量な Flash-Lite 系、といった役割分担です。試験対策としては、個々のバージョン番号を暗記するのではなく、「重い推論には上位モデル、高頻度・低遅延には Flash 系」という選び分けの原則を押さえてください。バージョンは更新され続けるため、番号そのものは問われにくく、選び分けの考え方が問われます。

Gemini の強みとして挙がるのは、multimodal を前提に設計されていること、大きな context window を扱えること、そして Google Cloud の各サービスや Google Workspace と統合されていることです。この統合は、後の 2-1 で扱う「comprehensive AI ecosystem(包括的な AI エコシステム)」という Google Cloud の強みの中核でもあります。

1-4-2 Gemma [1.4]

Gemma は、Gemini と同じ研究基盤から作られた open models(オープンモデル)のファミリーです。軽量で、モデルの重みが公開されており、自社の環境にダウンロードして動かせます。ノートパソコンや自社データセンター、エッジ端末でも動かせる規模のものが提供されています。

Gemma を選ぶ理由は主に三つです。第一に、データを外に出さずに完全に自社管理下で推論したい場合。第二に、モデルを深く改変したい場合。第三に、大量の推論を低コストで回したい場合です。Gemini が「Google が運用する強力な汎用モデル」であるのに対し、Gemma は「自分で持って回す軽量モデル」だと整理すると混同しません。

これは 2-1 で扱う「Google Cloud の open approach(オープンなアプローチ)」の具体例でもあります。自社の第一党モデルだけでなく、オープンモデル

とサードパーティモデルも同じプラットフォームから使えるようにしている、という姿勢の表れです。

1-4-3 Imagen [1.4]

Imagen は text-to-image、すなわちテキストの指示から画像を生成するモデルです。生成だけでなく、既存画像の編集、部分的な書き換え、背景の差し替えといった編集用途にも使われます。

ビジネスでのユースケースは、広告クリエイティブの案出し、EC の商品画像のバリエーション作成、社内資料やプレゼンの挿絵、キャンペーンビジュアルの多言語・多地域展開です。デザイナーを置き換えるというより、案の数を増やして選択と改善のサイクルを速める使い方が現実的です。

1-4-4 Veo [1.4]

Veo は text-to-video、すなわちテキストの指示から動画を生成するモデルです。静止画を起点にした生成にも対応します。ビジネスでのユースケースは、短尺の広告動画、商品紹介クリップ、研修や説明用の映像、SNS 向けコンテンツです。

モデル	modality	主な用途	選ぶ理由
Gemini	multimodal 入力 → テキスト出力	推論、要約、対話、コード生成、文書抽出	汎用性と統合。まず検討する既定の選択肢
Gemma	テキスト中心	自社環境での推論、改変、低コストの大量処理	open models。データを外に出さない、深く改変する
Imagen	テキスト → 画像	広告・商品・資料の画像生成と編集	画像を作る・直す必要がある

モデル	modality	主な用途	選ぶ理由
Veo	テキスト/画像 → 動画	短尺広告、商品紹介、研修映像	動画を作る必要がある

このほか、音声認識の Chirp、音楽生成の Lyria といったモデルも提供されています。台帳が主題に挙げているのは上の四つですので、四つの役割分担を確実にしたうえで、周辺の存在を知っている、という状態が試験には適しています。

誤答肢の切り方を確認します。「商品写真のバリエーションを作りたい」という場面で Gemini を選ぶのは誤りです。Gemini は画像を入力として理解できますが、画像そのものの生成は Imagen の役割です。「機密データを外部に出さずにオンプレミス環境で LLM を動かしたい」で Gemini を選ぶのも、この文脈では Gemma が素直な答えになります。「30秒の商品紹介動画を作りたい」で Imagen を選ぶのは、静止画と動画を取り違えているので誤りです。モダリティと役割で切る、というのがこの領域の解き方です。

1-4-5 第1章のまとめと、第2章への橋渡し [1.1/1.2/1.3/1.4]

第1章では、用語の入れ子(AI → machine learning → deep learning → generative AI)、foundation models と large language models と multimodal foundation models と diffusion models の区別、prompt engineering と prompt tuning の違い、supervised・unsupervised・reinforcement の三方式、machine learning lifecycle の五段階と Google Cloud tools、foundation model 選定の八つの軸、create・summarize・discover・automate のユースケース、data quality の六特性、structured と unstructured、labeled と unlabeled、そして Infrastructure から Applications までの五レイヤーと Google の四モデルを扱いました。

このドメインは採点対象の 30% を占め、四つのドメインの中で二番目に大きい比重です。ここで固めた語彙と対比が、第2章以降の製品選択、第3章の出力改善、第4章の戦略判断のすべての土台になります。特に foundation model 選定の軸と、prompt engineering から fine-tuning へ向かうコストの階段は、他の章で繰り返し参照される考え方です。

次章では、この地図の上に Google Cloud の具体的な製品群を置いていきます。どのレイヤーのどの課題に、どの製品が対応するのか。その対応関係を作ることが第2章の仕事になります。

1-5 補講: 取り違えやすい概念の整理と、データ種別の実務的な使われ方 [1.1/1.2/1.3/1.4]

1-5-1 生成AIで各データ種別がどう使われるか [1.2]

objective 1.2 は「various data types are used in gen AI」、すなわち各データ種別が生成AI の中で実際にどう使われるかを説明できることを求めています。1-2-2 で種別の定義を扱いましたので、ここでは使われ方の側から整理します。

テキストデータは、生成AI にとってもっとも中心的な入力です。事前学習の素材であり、prompt の本体であり、grounding で渡す根拠であり、出力そのものでもあります。ビジネス上の意味合いは、社内の文書資産がそのまま資源になるという点です。ただし表記ゆれや旧版の混在があると、consistency と relevance の欠如がそのまま誤答につながります。

画像データは、multimodal foundation models の入力として、また image generation の出力として使われます。入力としての用途は、写真からの状態判定、帳票からの項目抽出、図面の読み取りです。出力としての用途は広告や

資料の素材作成です。ビジネス上の意味合いは、これまで人間が目視でしか扱えなかった工程を、量をこなせる形に変えられることです。

音声データは、Speech-to-Text API でテキストへ変換してから扱うのが基本形です。コールセンターの通話、会議の録音、現場の音声メモが対象になります。テキスト化して初めて要約・検索・分析ができるようになるので、format の観点が効く典型例です。逆に Text-to-Speech API を使えば、生成したテキストを音声として返せます。

動画データは、Cloud Video Intelligence API による解析や、multimodal foundation models への直接入力、そして Veo による生成という三方向で使われます。監視映像からの異常検知、研修動画の要約、商品紹介クリップの生成が例です。

構造化データは、BigQuery 上の集計結果や、テーブルの行そのものを prompt に含めて説明させる、という使い方が中心です。自然言語の質問から SQL を生成させ、実行結果を言葉で解説させる流れは data analysis の定番です。ここで重要なのは、モデルに数値の計算そのものを任せるのではなく、集計はデータベースに任せ、モデルには問いの翻訳と結果の説明を任せる、という役割分担です。

データ種別	生成AI での使われ方	ビジネス上の意味合い	関係する Google Cloud の手段
テキスト	prompt、grounding の根拠、出力	文書資産がそのまま資源になる	Agent Search、Document AI API
画像	multimodal 入力、image generation の出力	目視工程を量産可能にする	Gemini、Imagen、Cloud Vision API

データ種別	生成AI での使われ方	ビジネス上の意味合い	関係する Google Cloud の手段
音声	テキスト化してから分析、音声としての出力	通話・会議が検索・分析の対象になる	Speech-to-Text API、Text-to-Speech API
動画	解析、multimodal 入力、video generation	映像資産の要約・監視・制作	Gemini、Veo、Cloud Video Intelligence API
構造化	集計結果の説明、SQL 生成の対象	既存 BI と自然言語の橋渡し	BigQuery、BigQuery ML

1-5-2 取り違えやすい用語の対比まとめ [1.1/1.3/1.4]

ここまでに出た用語のうち、選択肢で並べられたときに迷いやすい組み合わせを一覧にします。試験直前にはこの表だけを見返せば足りる、という密度でまとめます。

迷いやすい組み合わせ	決め手となる違い
artificial intelligence と machine learning	AI が外側の傘、machine learning はその内側の手法群
generative AI と predictive AI	新しい成果物を作るか、ラベルや数値を当てるか
foundation model と large language models	前者が上位概念、後者は言語を主対象とする一種
multimodal foundation models と単一モダリティのモデル	複数の入出力種別を同時に扱えるかどうか
diffusion models と LLM	ノイズ除去で画像を作るか、次の token を予測するか

迷いやすい組み合わせ	決め手となる違い
prompt engineering と prompt tuning	指示文の工夫か、学習可能な付加ベクトルの学習か
prompt tuning と fine-tuning	モデル本体の重みを変えないか、変えるか
supervised と unsupervised	正解ラベルがあるか、データ自体の構造から学ぶか
unsupervised と reinforcement	構造を見つけるか、報酬を最大化する行動を学ぶか
structured data と unstructured data	行・列・型のスキーマがあるか、ないか
labeled data と unlabeled data	各データ点に正解が付いているか、いないか
structured data と labeled data	形式の話か、正解の有無の話か。別の軸である
context window と parameters	一度に渡せる token 量か、モデルの重みの数か
Gemini と Gemma	Google が運用する汎用 multimodal か、自分で持って回す open models か
Imagen と Veo	画像を作るか、動画を作るか
Models 層と Platforms 層	モデルそのものか、モデルを業務で使う土台か
Agents 層と Applications 層	自律的に手順を組んで実行する仕組みか、利用者が触る画面か

1-5-3 場面問題の解き方の型 [1.1/1.2/1.3/1.4]

Leader 級の問いは、多くが場面設定つきで出されます。「ある企業のある部門が、こういう状況で、こうしたい。最適な選択はどれか」という形です。この型に

対しては、次の順で読むと安定します。

第一に、求められている出力の種類を見ます。文章か、画像か、動画か、コードか、分類結果か。ここで modality が決まり、Gemini・Imagen・Veo のどれか、あるいは predictive AI の領域かが分かれます。

第二に、根拠となるデータがどこにあるかを見ます。モデルが事前学習で知っているはずの一般知識で足りるのか、社内文書が要るのか、最新の外部情報が要るのか。社内文書が要るなら grounding や RAG、最新の外部情報が要るなら Grounding with Google Search が候補になります。この判断は第3章で詳しく扱います。

第三に、制約を見ます。データを外に出せないのか、遅延が厳しいのか、予算が限られるのか、規制があるのか。制約は候補を減らすために使います。

第四に、選択肢のうち「業務要件から出発していないもの」を落とします。「最大のモデルを使う」「最新のモデルを使う」「すべてを fine-tuning する」といった、要件と無関係に強い手段を推す選択肢は、Leader 級の試験ではほぼ誤答です。コストと可逆性の低い手段を最初に選ぶ選択肢も同様です。

この四段階は、第2章の製品選択、第3章の技法選択、第4章の戦略判断でも同じように使えます。覚えるのは順番だけです。

1-5-4 レイヤーごとの「買うか作るか」の判断 [1.3]

1-3 で見た五レイヤーは、投資判断の地図としても使えます。下の層ほど参入障壁が高く差別化しにくく、上の層ほど自社固有の知識が効きます。

Infrastructure と Models は、原則として買う層です。ここに自前で投資して他社を上回れる企業はごく限られます。cloud computing を通じて必要なだけ使うのが合理的です。

Platforms は、統一して使う層です。自社で同等のものを作ることは可能ですが、権限管理・監視・課金・評価といった機能を自作して維持し続ける負担に見合う成果は、たいていの企業では出ません。

Agents と Applications が、自社の差別化が生まれる層です。自社の業務手順、自社のデータ、自社の顧客理解が入るからです。ここに人と時間を集中させ、下の層は買う。これが Leader として説明すべき投資配分の骨格になります。

ただし、いきなり Agents から始めるのは推奨されません。まず Applications 層の既製品で価値を体験し、次に grounding で自社データをつなぎ、それから Agents へ進む。可逆で低コストな手段から段階的に上げていく、という第1章を通じた原則がここでも当てはまります。

サンプルはここまでです

続き(第2章〜巻末)は、GCP GenAI Leader 問題集のご購入で、頻出度別ノート・暗記用ノートと合わせて3点すべてダウンロードできます。

GCP GenAI Leader 学習用テキスト(無料サンプル)

版

2026年7月版(Google Cloud 公式 Generative AI Leader Exam Guide（2026-09-07 取得）準拠)

発行

IT資格道場(https://www.shikaku-doujou.com)

お問い合わせ

info@shikaku-doujou.com