IT資格道場

AWS AIF-C01 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)

AWS 公式 Exam Guide(AIF-C01・バージョン 1.1・2026年4月30日)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

3点セットの中の位置づけ

このシリーズは3つのテキストで構成されています。

- **学習用**: これだけで問題が解けるようになる本編
- **頻出度別**: 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」に橋渡しします。試験直前の最終チェックにも
- **暗記用**: 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: これだけで問題が解けるようになる本編です

はじめに —— この教材の使い方

このテキストは、**Amazon Web Services（AWS）**が実施する **AWS Certified AI Practitioner（試験コード AIF-C01）**の合格を目的に作られています。

この試験が問うのは、モデルを作れることでも、コードを書けることでもありません。問われるのは、**AI・機械学習・生成AIの概念と、それを AWS の上でどう使い、どこに気をつけるかを、業務の場面で判断できること**です。公式の試験ガイドも、モデルの開発やコーディング、ハイパーパラメータ調整、パイプライン構築、数学的分析は「対象外の職務」と明記しています。

本書は **AWS が公開している試験ガイド（Exam Guide）バージョン 1.1（2026年4月30日公開）のドメイン・タスク・到達目標を体系化した教材**であり、実際に出題された問題を再現したものではありません。**本書は AWS 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。**

AWS の実務経験は要りません。 公式ガイドが想定するのは「AWS 上の AI/ML 技術に触れて6か月まで」の人で、AI/ML を作る側ではなく使う側です。EC2・S3・Lambda・IAM・共有責任モデル・料金モデルといった AWS の基礎は、本文の中で出てきた場所で必ず一言の説明を添えました。

試験の基本情報

項目	内容
試験名	AWS Certified AI Practitioner（AIF-C01）
実施	Amazon Web Services（Pearson VUE のテストセンター、またはオンライン監督試験）
試験時間	**90分**
問題数	**65問**（うち採点対象 50問・非採点 15問。どれが非採点かは受験者には分かりません）
出題形式	**択一（4肢1正解）／複数選択（5肢以上から2つ以上）／並べ替え（3〜5項目）／対応付け（3〜7組）**
合格ライン	**スケールスコア 100〜1,000 のうち 700**
採点方式	全体で合否を決める補償型。分野ごとの足切りはありません
受験料	**100 USD**（日本円の受験料は AWS 公式の申込画面で確認してください）

項目	内容
受験言語	日本語で受験できます（英語ほか計12言語）

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、AWS 公式サイトで最新の情報をご確認ください。

試験ガイドの版について —— 本書は v1.1 に沿っています

この試験は 2026年4月30日に試験ガイドが **バージョン 1.1** に改訂され、約1か月後から試験問題に反映されています。追加されたのは、**エージェント型AI（agentic AI）・マルチエージェント・MCP（Model Context Protocol）・Amazon Bedrock AgentCore・Kiro・Strands Agents・Amazon Quick・コンテキストエンジニアリング・トークン課金と費用/性能・モデル蒸留・LLM-as-a-judge・幻覚検知とグラウンディング** など、生成AIを「動かす」段階の論点です。古い教材にはこれらがありません。本書は v1.1 の到達目標をすべて章立てに組み込んでいます。

出題範囲の全体像 —— 5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	AIとMLの基礎	**20%**	第1章
2	生成AI（GenAI）の基礎	**24%**	第2章
3	基盤モデルの応用	**28%**	第3章
4	責任あるAIのためのガイドライン	**14%**	第4章

ドメイン	分野	採点対象に占める割合（公式）	本書
5	AIソリューションのセキュリティ・コンプライアンス・ガバナンス	**14%**	第5章

本書の章の並びは、試験ガイド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、試験ガイドの Task Statement の番号です。公式ガイドを手元に置いて対照しながら読めます。AWS はドメインごとの割合だけを公表しており、Task 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。

時間配分 —— 1問あたり約80秒

90分で65問ということは、1問あたり約1分20秒です。並べ替え・対応付けは択一より時間がかかるので、択一を1分以内で片付けて貯金を作る配分が安全です。オンライン受験でも手元で調べる時間は実質ありません。目指すのは、**サービス名や用語を見た瞬間に「どのドメインの、どの場面の話か」が反射で出る状態**です。

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: 作る話か、使う話か	対象外の職務との境目	この試験は「使う側」。モデルを訓練・実装する手順の細部を答えさせる肢は、たいてい誤答です

問い	何を切り分けるか	分かれ方
問い2: どの段階の話か	AI/ML パイプラインの位置	データ準備／学習／評価／デプロイ／運用監視 —— 同じサービス名でも段階が違えば答えが違います
問い3: 何を根拠に選ぶか	費用・性能・説明可能性・規制	従来型 ML と基盤モデル、RAG とファインチューニング、オンデマンドとプロビジョンド —— 選択肢は必ずトレードオフで問われます

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、5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4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 AIとMLの基礎(ドメイン1・採点対象の20%)

この章は、AIF-C01 の土台です。採点対象の20%を占め、ここで固めた用語がドメイン2以降(生成AI、基盤モデルの応用、責任あるAI、セキュリティとガバナンス)の設問文にそのまま出てきます。用語を取り違えたまま先へ進むと、後半の設問で「何を聞かれているのか」から迷うことになります。

出題の型は三つです。①用語の定義を入れ替えた選択肢から正しいものを選ぶ、②業務の場面を示して「その課題にAI/MLは適切か、適切ならどの手法か」を選ぶ、③AI/MLの工程を示して「その段階に当たるAWSサービスはどれか」を選ぶ。どれも、**一言定義と、似た用語との違いを対で覚える**ことで解けます。

AIF-C01 の受験者像は「AI/MLモデルを自分で実装する人」ではありません。試験ガイドが対象外と明記しているのは、モデルやアルゴリズムの開発・コーディング、データエンジニアリングや特徴量エンジニアリングの実装、ハイパーパラメータのチューニングや最適化、AI/MLパイプラインやインフラの構築と展開、AI/MLモデルの数学的・統計的な分析、AI/MLシステムのセキュリティやコンプライアンスの実装、ガバナンスのフレームワークやポリシーの策定です。**手を動かして作る側の詳細は問われず、選ぶ側・使う側としての判断が問われる**と考えてください。

試験の形式は、採点対象50問と採点対象外15問を合わせた**65問**、**90分**、合格ラインは**700点**(100〜1000点のスケールドスコア)です。設問の形式は多肢選択(multiple choice)、複数選択(multiple response)、順序付け(ordering)、組み合わせ(matching)の4種類があります。順序付けはパイプラインの工程順、組み合わせはサービスと用途の対応で出やすく、どちらも本章の内容が直接の材料になります。

1-1 AIの基本的な概念と用語 [1.1]

1-1-1 AI・ML・深層学習・ニューラルネットワークの入れ子関係 [1.1]

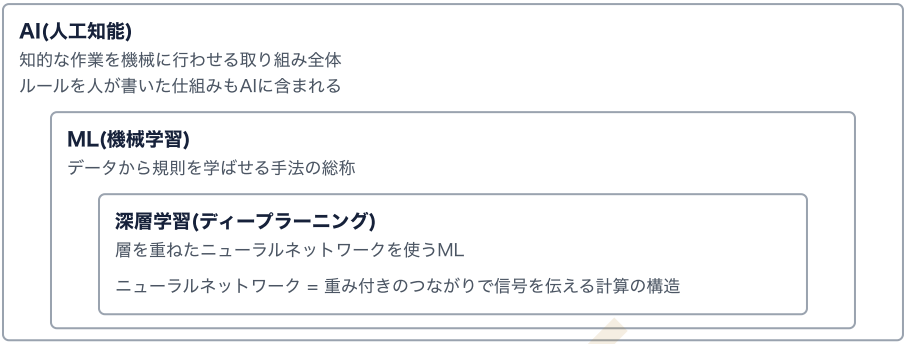
いちばん最初に固めるのは、四つの言葉の包含関係です。**入れ子**になっています。

> **AI ⊃ ML(機械学習) ⊃ 深層学習(ディープラーニング)** ニューラルネットワークは深層学習が使う**構造の名前**

一言定義を並べます。

- **AI(人工知能)**……人間の知的な作業(見る、聞く、読む、判断する、作る)を機械に行わせる取り組み**全体**を指す大きな枠。ルールを人が書き下した仕組みもAIに含みます。
- **ML(機械学習)**……AIの一分野で、**規則を人が書くのではなく、データから規則を学ばせる**手法の総称。
- **深層学習(ディープラーニング)**……MLの一分野で、**層を深く重ねたニューラルネットワーク**を使う手法。画像・音声・自然言語のように、人が特徴を書き出しにくいデータで強みが出ます。
- **ニューラルネットワーク**……入力層・隠れ層・出力層からなる、**重み付きのつながりで信号を伝える計算の構造**。隠れ層を多数重ねたものが深層学習です。

AI・ML・深層学習・ニューラルネットワークの入れ子関係



「MLはAIより広い概念である」→ 逆。AIの中にMLがある。
「深層学習はMLとは別系統の技術で、重なりは無い」→ 深層学習はMLの一部。
「ニューラルネットワークは深層学習の上位概念である」→ 構造の名前。

図 1-1　AI・ML・深層学習・ニューラルネットワークの入れ子関係

よく出る取り違えを表にします。

誤った説明	正しくは
MLはAIより広い概念である	逆。**AIの中にML**がある
深層学習はMLとは別系統の技術で、重なりは無い	**深層学習はMLの一部**
ニューラルネットワークは深層学習の上位概念である	ニューラルネットワークは**構造の名前**。層を深く重ねた使い方が深層学習
AIはすべて機械学習で作られている	ルールを人が書いた仕組みもAIに含まれる
深層学習は少量のデータで従来手法より必ず高精度になる	深層学習は**大量のデータと計算資源**を前提にすることが多い

使い分けの目安も押さえます。表形式(テーブル)の数千件程度の業務データから金額を予測するなら、深層学習を持ち出すより**従来型のML(勾配ブースティングや回帰)で十分**なことが多く、学習の速さ・説明のしやすさで有利で

す。逆に、傷の写真から不良品を見分ける、通話の音声を文字にする、といった課題は**深層学習の領域**です。「新しい技術だから深層学習を選ぶ」は、AIF-C01 では誤答の典型です。

1-1-2 モデル・アルゴリズム・学習と推論 [1.1]

この4語は、選択肢を入れ替えるだけで設問が作れるので頻出します。

- **アルゴリズム**……データから規則を見つけ出すために**あらかじめ用意された計算の手順**。モデルを作るための「道具の側」です。
- **モデル**……学習の結果として得られた、**入力から出力を導くための規則(パラメータ)の集まり**。道具ではなく**成果物**です。
- **学習(トレーニング)**……データを読み込ませて、**モデル内部の値(重み・パラメータ)を調整していく工程**。計算資源を多く使い、時間がかかります。
- **推論(インファレンス)**……学習を終えたモデルに**新しい入力を与えて出力を受け取る工程**。日々の業務で動くのはこちらです。

判別の軸は「内部の値が変わるかどうか」です。 学習では変わり、推論では変わりません。「推論のたびにモデルが少しずつ賢くなる」は誤りで、賢くするには**再学習**という別の工程が要ります。

場面	どの工程か
部門ごとに散らばった記録を1つの表にそろえ、項目名を統一する	学習の前の**データ準備**(モデルはまだ無い)
過去3年分の受注実績を読み込ませ、パラメータを調整する	**学習**
完成したモデルに今月の実績を渡し、翌月の見込みを受け取る	**推論**

場面	どの工程か
予測と実績の差を毎月集計し、精度の推移を報告する	運用後の**モデル監視**
精度が落ちてきたので、直近データを足してもう一度学習する	**再学習**

コストの出方も違います。**学習は一時的に大きな計算資源を使う「山」型、推論は使われるたびに少しずつ積み上がる「流れ」型**です。生成AIでは推論が使用量課金(トークン課金)で積み上がるため、長期の総コストでは推論側が支配的になることがあります。「学習が終われば以後の費用はかからない」は誤りです。

1-1-3 コンピュータビジョンと自然言語処理(NLP) [1.1]

AIの応用分野には名前が付いており、設問では「この課題はどの分野か」を問われます。

- **コンピュータビジョン**……**画像や動画の中身を機械に理解させる**分野。物体検出、画像分類、顔の検出、文字の読み取り(OCR)、外観検査などが含まれます。
- **自然言語処理(NLP)**……**人が書いた・話した言葉を機械に扱わせる**分野。文章の分類、感情分析(ポジティブ/ネガティブ)、固有表現の抽出(人名・組織名・日付)、要約、翻訳、質問応答が含まれます。
- **音声認識(自動音声認識)**……**音声を文字に変換する**技術。文字にした後の処理はNLPの領域に移ります。
- **音声合成(テキスト読み上げ)**……逆向きに、**テキストを音声に変換する**技術。

取り違えの注意点。「コールセンターの通話を分析したい」という課題は、まず**音声認識で文字にし(Amazon Transcribe)、次にNLPで感情や話題を分析する(Amazon Comprehend)**という二段構えになります。片方だけで完結すると考えると誤答します。同様に「請求書の画像から金額と日付を取り出す」は、**画像から文字を取り出す工程(コンピュータビジョン/OCR)**が先で、その後に項目の判別が入ります。

1-1-4 大規模言語モデル(LLM)・生成AI(GenAI)・基盤モデル(FM) [1.1]

- **生成AI(GenAI)**……学習した内容をもとに、**文章・画像・音声・コードといった新しい成果物を作り出す**AIの総称。
- **基盤モデル(FM: Foundation Model)**……**大量かつ多様なデータで事前学習され、幅広い用途に転用できる大規模なモデル。**1つのモデルで要約・翻訳・分類・作文など多数のタスクをこなせるのが特徴です。
- **大規模言語モデル(LLM)**……FMのうち、**主にテキストを扱うもの。**次に来る語句を確率的に選んで文章を組み立てます。
- **マルチモーダルモデル**……テキストに加えて**画像・音声など複数の種類の入力**を扱えるFM。

関係を一言で言えば、**生成AI ⊃ FM ⊃ LLM** の向きで包含が成り立ちます(厳密には生成AIにはFMを使わない古典的な生成手法も含まれますが、試験の文脈ではこの入れ子で足ります)。

生成AIの性質で押さえるのは三つです。

1. **同じ指示でも出力が毎回同じとは限らない**(確率的に生成するため)。
2. **事実と異なる内容をもっともらしく作り出すことがある**(ハルシネーション/幻覚)。

3. **学習した時点より新しい情報は知らない**(知識のカットオフ)。

裏返すと、「生成AIは出力の正しさを内部で検証してから返す」「同じ入力には常に同じ出力を返す」は**誤り**です。だからこそ、社内文書を根拠に答えさせる**グラウンディング**(RAG=検索拡張生成やナレッジベースの利用)と、人の確認を挟む運用設計が必要になります。

トークンという単位も、ここで押さえます。LLMは文章を**トークン**という細かい単位に分けて処理し、多くのサービスが**入力トークン数と出力トークン数に応じた従量課金(トークン課金)**を採ります。この課金モデルの結果、**プロンプトを長くするほど、また長い回答を出させるほど費用が上がり、応答時間も伸びます**。費用と性能の両方に効くため、必要な文脈だけを渡す設計(コンテキストエンジニアリング)が実務の要点になります。

1-1-5 エージェント型AI(agentic AI)とマルチエージェント [1.1]

v1.1 で強調された領域です。定義を正確に。

- **エージェント型AI(agentic AI)**……与えられた目標に対して、**モデル自身が手順を計画し、外部のツールやAPIを呼び出し、結果を見て次の行動を決める**という自律的な動きをするAIの形。単発の質問応答と違い、**複数ステップの行動と外部システムへの働きかけ**を伴います。
- **AIエージェント**……その仕組みの実体。一般に「モデル(頭脳)+ツール(手足)+メモリ(記憶)+オーケストレーション(手順の制御)」で構成されます。
- **マルチエージェント**……役割の異なる複数のエージェントが**分担・連携**して1つの目標に当たる構成。調査役・執筆役・検証役に分けるといった形です。
- **MCP(Model Context Protocol)**……**モデルと外部のデータソースやツールをつなぐための共通の取り決め(オープンな標準)**。サービスごとに独

自の接続を作らず、同じ約束事でツールをつなげるようにするものです。

AWS側の対応する要素も、名前と役割だけ押さえます。

名称	位置づけ
Amazon Bedrock AgentCore	エージェントを本番運用するための基盤。実行環境、ツール接続、記憶、監視をまとめて提供する。**AgentCore Identity** はエージェントが他システムへアクセスする際の認証・認可を扱い、**AgentCore Policy** はエージェントの行動に対する制約(何をしてよいか)を扱う
Strands Agents	エージェントを組み立てるための**オープンソースのSDK**。モデル・ツール・プロンプトを指定してエージェントを構築する開発者向けの道具
Kiro	AI を前提とした**開発の道具(IDE)**。仕様から実装へつなげる形で開発作業を支援する。エージェントの実行基盤ではなく、**作る側の環境**

取り違えの注意。「エージェント型AIは、より賢い大きなLLMのこと」は誤りです。**モデルの大きさの話ではなく、自律的に計画し外部を操作するという振る舞いの話**です。また「エージェントを使えば人の承認は不要になる」も誤りで、外部システムを操作する以上、**権限の最小化と行動の記録(監査ログ)、重要な操作への人の承認**はむしろ強く求められます。

1-1-6 バイアス・公平性・フィット(過学習と未学習) [1.1]

- **バイアス(偏り)**……モデルの出力が、**特定の集団や特定の傾向に系統的に偏る**こと。学習データの偏り、ラベル付けの基準の揺れ、特徴量の選び

方などが原因になります。

- **公平性(フェアネス)**……**属性(性別・年齢・地域など)によって不当な扱いの差が生じていないか**という観点。バイアスは症状、公平性はそれを測り是正する枠組み、と対で覚えます。
- **フィット**……モデルがデータにどれだけ当てはまっているか。**過学習(オーバーフィッティング)**と**未学習(アンダーフィッティング)**の両極があります。

状態	症状	典型的な対処
過学習(オーバーフィッティング)	学習データでは高精度なのに、**未知のデータで精度が落ちる**。データの雑音まで覚え込んでいる	データを増やす、特徴量を減らす、正則化、早期打ち切り
未学習(アンダーフィッティング)	**学習データでも精度が出ない**。モデルが単純すぎる、学習が足りない	より表現力のあるモデルにする、学習を長くする、特徴量を足す
良いフィット	学習データと未知データの両方で同程度に良い	—

判別の合言葉。「学習では良いが本番で悪い」→**過学習**。「学習でも本番でも悪い」→**未学習**。この一行で多くの設問が解けます。

バイアスの現れ方も場面で覚えます。過去の採用実績で学習した書類選考モデルが、過去の偏った採用傾向をそのまま再現してしまう。首都圏の店舗だけの記録で作った需要予測を地方に広げると外れる。どちらも**学習データが対象の母集団を覆っていない**ことが原因で、件数を増やすだけでは直りません。**同じ偏りを持つ記録を積み上げても、偏りは打ち消されない**からです。

なお公平性・バイアスの是正や責任あるAIの枠組み(有害性、説明可能性、透明性など)は、本試験ではドメイン4で深く問われます。ここでは**用語の定義と、**

症状の見分けまでを確実にしておきます。

1-1-7 AI・ML・GenAI・深層学習・エージェント型AIの類似点と相違点 [1.1]

5つを一枚の表で比べます。**類似点は「データから学ぶ」「確率的に振る舞う」「品質の監視が要る」、相違点は「範囲」と「出力の形」と「自律性」**です。

用語	何を指すか	出力の形	位置づけ
AI	知的作業を機械に行わせる取り組み全体	分野により様々	いちばん外側の枠
ML	データから規則を学ぶ手法の総称	数値・分類ラベルなど**決まった形式**	AIの一部
深層学習	層を重ねたニューラルネットワークを使うML	ML同様。画像・音声・言語に強い	MLの一部
GenAI	新しい成果物を作り出すAI	文章・画像・音声など**形の定まらない出力**	深層学習の応用
エージェント型AI	目標に対し自ら計画しツールを使うAI	**行動と、その結果としての成果**	GenAIの応用形

相違点の一言まとめ。

- **MLとGenAIの違い**……MLは**予測・分類**(既存の答えを当てる)、GenAIは**生成**(新しいものを作る)。
- **GenAIとエージェント型AIの違い**……GenAIは**求められた出力を返すところまで**、エージェント型AIは**外部のツールを呼び、複数の手順を自分で進める**。

- **深層学習とGenAIの違い**……深層学習は**手法**の名前、GenAIは**目的(何を出すか)**の名前。深層学習で分類器を作れば、それはGenAIではありません。

共通点として問われるもの。どれも**データに依存し、出力に誤りが起こりうるため、評価と監視が必要**という点は共通です。「エージェント型AIは自律的なので人の監督は不要」「生成AIは学習データに依存しない」といった選択肢は、この共通点に反するので切れます。

1-1-8 推論の種類(バッチ・リアルタイム・非同期・サーバーレス) [1.1]

推論の実行形態は、**要件(応答速度・件数・入力サイズ・コスト)からどれを選ぶか**が問われます。4種類を対比表で覚えます。

種類	動き方	向く場面	コストと待ち時間
バッチ推論	大量の入力をまとめて処理し、結果をS3などにまとめて出す。**常時起動のエンドポイントを持たない**	夜間に全顧客の解約スコアを一括算出、月次で全商品の需要を予測	**単位あたり最安**。結果が出るまで待つ(即時性は無い)
リアルタイム推論	常時起動のエンドポイントに1件ずつ要求を送り、**ミリ秒〜秒**で応答を得る	決済時の不正検知、サイト表示時のレコメンデーション、チャットの応答	**常時起動のぶん高コスト**。待ち時間は最短
非同期推論	要求を**キューに入れて順に処理**し、完了後に結果を通知・保存する。処理中は待たされない	入力サイズが大きい(長い動画・大きな画像・長文書)、処理に数分かかる、要求が波打つ	中間。**待ち時間は数分許容**、処理が無いときはゼロまで縮小できる

種類	動き方	向く場面	コストと待ち時間
サーバーレス推論	サーバーの管理不要で、**要求が来たときだけ立ち上がる**。使わない時間の課金が発生しない	**要求が間欠的・予測しにくい**、社内向けで夜間は使われない	待機コストが不要。**初回起動の遅れ(コールドスタート)**が出る

選択の判断軸を3つだけ。

1. **即時に返す必要があるか** → はい:リアルタイム。いいえ:バッチか非同期。
2. **要求が常時来るか、たまにしか来ないか** → たまに:サーバーレス(待機コストを払わない)。
3. **1件の入力が大きい、または処理が長いか** → はい:非同期(タイムアウトを避け、キューで受ける)。

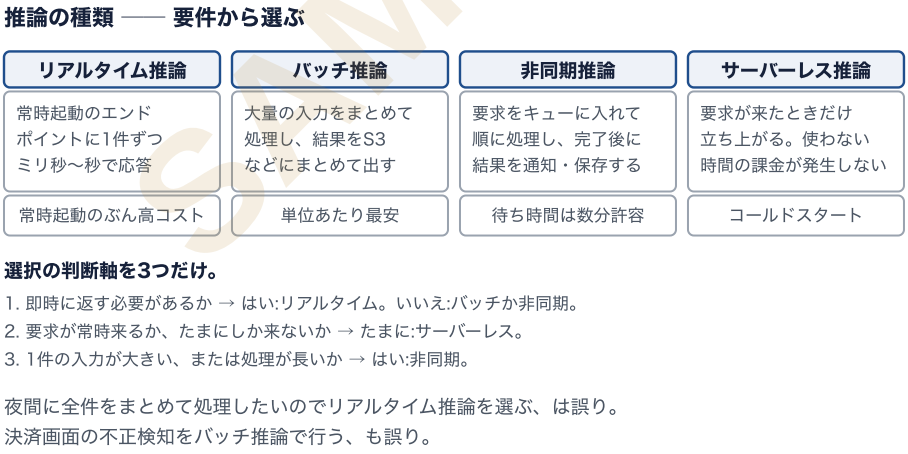


図 1-2　推論の種類 —— 要件から選ぶ4つの実行形態

よく出る取り違え。

誤った選択	なぜ違うか
夜間に全件をまとめて処理したいのでリアルタイム推論を選ぶ	即時性が不要なのに常時起動の費用を払うことになる。**バッチ推論**が適切
1日に数回しか使わない社内ツールに常時起動のエンドポイントを置く	待機時間の課金が無駄。**サーバーレス推論**が適切
30分かかる長時間処理をリアルタイム推論で受ける	応答の待ち時間の上限を超える。**非同期推論**が適切
決済画面の不正検知をバッチ推論で行う	判定が取引に間に合わない。**リアルタイム推論**が適切
コールドスタートが許容できない業務にサーバーレス推論を選ぶ	初回の遅れが要件に反する。**リアルタイム推論**(プロビジョンド)が適切

1-1-9 AIモデルで扱うデータの種類 [1.1]

データの分け方は**2つの軸**があります。混ぜないことが要点です。

軸1:ラベルの有無

- **ラベル付きデータ**……各データに**正解(答え)が付いている**。例:「この画像は不良品」「この取引は不正」「この顧客は解約した」。**教師あり学習**の材料になります。
- **ラベルなしデータ**……正解が付いていない生のデータ。**教師なし学習**(クラスタリングなど)や、FMの事前学習で使われます。

ラベル付けには**人手のコストと時間**がかかります。「ラベル付きデータが少ししか無い」という制約は設問でよく登場し、そのときの選択肢は**ラベルなしデータで教師なし学習を行う**、**事前学習済みモデルを流用する**、**ラベル付けの工数を確保する**のいずれかになります。「ラベルが無いまま教師あり学習を行う」は成立しません。

軸2:データの形式

形式	中身	例	相性のよい扱い
表形式(テーブル)	行と列が決まった表	売上明細、顧客マスタ、センサーの集計値	従来型MLの回帰・分類
時系列	**時刻の順序に意味があるデータ**	日次売上、電力使用量、株価、機器の稼働ログ	需要予測、異常検知
画像	写真・スキャン・動画のフレーム	検査画像、レントゲン、防犯カメラ	コンピュータビジョン
テキスト	自由記述の文章	問い合わせメール、レビュー、契約書	NLP、LLM
音声	録音された音	通話録音、会議の録音	音声認識

構造化・非構造化は、この形式をまとめた呼び方です。

- **構造化データ**……行と列の形が決まり、項目ごとの意味が定義されているデータ。**そのまま集計・絞り込みに使えます**。表形式のデータが代表です。
- **非構造化データ**……行と列の枠に収まらないデータ。画像・テキスト・音声・動画が該当します。**中身を取り出す工程を挟んでから活用します**。
- **半構造化データ**……JSONやXML、ログのように、**厳密な表ではないが構造の手がかりを持つ**データ。

時系列データの注意点は独立して問われます。時系列は**順序と時刻の間隔に意味がある**ため、①学習と評価を**時間で分ける**(未来のデータで学習して過去を予測しない)、②季節性・トレンド・周期を考慮する、③欠測した時刻の扱

いを決める、という配慮が要ります。「時系列データも他の表形式データと同様に無作為に分割してよい」は誤りです。

取り違えの整理。

誤った説明	正しくは
画像や音声も表にそのまま格納できるので構造化データである	画像・音声は**非構造化データ**
表形式のデータはAIには使えない	需要予測も解約予測も**表形式データ**から作る
ラベルなしデータは価値が無い	クラスタリングや異常検知、FMの事前学習で使う
件数を増やせばラベルの品質の問題は消える	基準の揺れは**そのまま学習に取り込まれる**。件数では直らない

データの品質は、種類とは別の軸で問われます。点検の観点は四つです。

観点	症状の例	影響
欠損	住所の欄が空のままの取引先が2割ある	その項目を使う予測の精度が落ちる
重複	同じ顧客が別の番号で二重登録されている	その傾向が過大に学習される
一貫性	部門ごとに日付の書式や取引先名の表記が違う	同じ相手が別の相手として集計される
鮮度・粒度	商品マスタが半年前のまま。在庫数が月末しか更新されない	現在の傾向を反映できない

押さえる原則は三つ。①件数を増やしても、誤りや偏りは打ち消されません(同じ偏りが積み上がるだけです)。②**ラベルの基準が担当者ごとに揺れていれば、そのばらつきがそのまま学習に取り込まれます。③材料が崩れているなら、アルゴリズムの入れ替えでは直りません。**欠損と重複を洗い出し、記録の取り方から整えるのが先です。「精度が出ないので別の手法に替える」は、原因がデータ側にあるときは的外れな対処になります。

1-1-10 AI/ML学習の種類(教師あり・教師なし・強化学習) [1.1]

学習の種類は、**材料(ラベルの有無)と目的**で見分けます。

種類	材料	目的	代表的な手法	業務例
教師あり学習	**ラベル付きデータ**	正解を当てる	回帰、分類	需要予測、解約予測、不正検知、画像の良品/不良品判定
教師なし学習	**ラベルなしデータ**	構造を見つける	クラスタリング、次元削減、異常検知	顧客セグメント分け、異常な取引の洗い出し
強化学習	**環境との試行錯誤と報酬**	累積の報酬を最大にする行動方針を学ぶ	方策学習	ロボット制御、在庫の発注方策、ゲームAI、広告配信の最適化

補助的に出てくるものも一言で。

- **半教師あり学習**……**少量のラベル付きと大量のラベルなし**を組み合わせる。ラベル付けの費用を抑えたいときの選択肢。

- **自己教師あり学習**……データ自身から擬似的な正解を作って学ぶ。**FMの事前学習**はこの形です。
- **転移学習**……**別の課題で学習済みのモデルを土台にして、少ないデータで目的の課題に合わせる**。ファインチューニングはこの一種です。
- **RLHF(人間のフィードバックによる強化学習)**……人が付けた好みの順位を報酬として、モデルの応答を望ましい方向へ調整する手法。強化学習の応用で、LLMの調整に使われます。

見分けの合言葉。

- 「過去の**答え**がある」→**教師あり学習**
- 「答えは無いが、**似たもの同士をまとめたい・普段と違うものを見つけたい**」→**教師なし学習**
- 「**行動して結果を受け取り、繰り返して上達させたい**」→**強化学習**

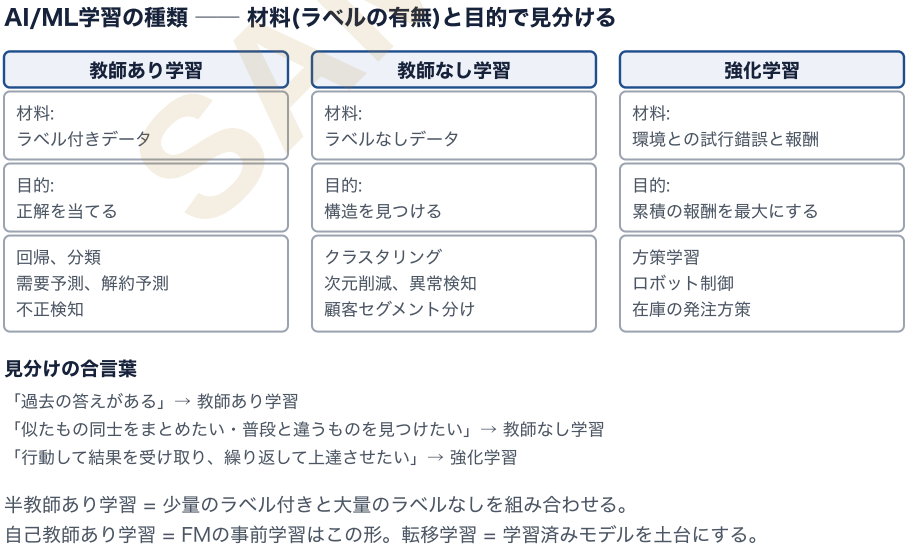


図 1-3　AI/ML学習の種類 —— 材料(ラベルの有無)と目的で見分ける

よく出る取り違え。

場面	誤答	正解
過去の解約実績(解約した/しなかった)から解約しそうな顧客を見つける	教師なし学習	**教師あり学習(分類)**
ラベルの無い購買履歴から顧客をいくつかのグループに分ける	教師あり学習	**教師なし学習(クラスタリング)**
倉庫ロボットが試行を繰り返して最短経路を身につける	教師あり学習	**強化学習**
ラベル付けの費用が高いので一部だけラベルを付けて活用したい	強化学習	**半教師あり学習**
汎用のFMを自社の文体に合わせたい	強化学習	**ファインチューニング(転移学習)**

1-1-11 用語カード(一言定義の総ざらい) [1.1]

設問文に出た瞬間に意味が浮かぶよう、ここまでの用語を一言で並べ直します。**この表の左から右を言えるか、右から左を当てられるか**の両方向で確認してください。

用語	一言で
AI	知的な作業を機械に行わせる取り組み全体
ML(機械学習)	データから規則を学ばせる手法の総称。AIの一部

用語	一言で
深層学習	層を重ねたニューラルネットワークを使うML
ニューラルネットワーク	重み付きのつながりで信号を伝える計算の構造
コンピュータビジョン	画像・動画の中身を理解させる分野
自然言語処理(NLP)	人の言葉を機械に扱わせる分野
モデル	学習の結果として得られた、入力から出力を導く規則の集まり
アルゴリズム	規則を見つけ出すための計算の手順
学習	データを使ってモデル内部の値を調整する工程
推論	学習済みモデルに新しい入力を与えて出力を得る工程
バイアス	出力が特定の集団や傾向へ系統的に偏ること
公平性	属性による不当な差が出ていないかを測り是正する観点
フィット	データへの当てはまり具合。過学習と未学習の両極がある
大規模言語モデル(LLM)	主にテキストを扱う大規模な基盤モデル
生成AI(GenAI)	新しい文章・画像・音声などを作り出すAI
エージェント型AI	目標に対し自ら計画し、ツールを呼んで手順を進めるAI

用語	一言で
基盤モデル(FM)	大量多様なデータで事前学習され、幅広い用途に転用できるモデル
トークン	LLMが文章を処理する細かい単位。課金と処理量の基準になる
ハルシネーション	事実と異なる内容をもっともらしく生成すること
グラウンディング	信頼できる根拠に基づいて回答させ、ハルシネーションを抑えること
コンテキストエンジニアリング	モデルに渡す文脈(指示・参考資料・履歴)を、必要なものだけ適切な形で設計すること
MCP(Model Context Protocol)	モデルと外部のデータ・ツールをつなぐ共通の取り決め
マルチエージェント	役割の違う複数のエージェントが分担・連携する構成
モデル蒸留	大きなモデルの振る舞いを小さなモデルに学習させ、推論を安く速くする手法
LLM-as-a-judge	生成物の良し悪しを別のモデルに判定させる評価方法

コンテキストエンジニアリングの位置づけを補足します。LLMが一度に読める文脈の量には上限があり、**渡す量が増えるほどトークン課金の費用と応答時間が増えます**。だから「関係しそうな資料を全部貼る」のではなく、**質問に答えるのに必要な情報だけを、必要な順序と形で渡す**設計が要ります。RAG(ナレッジベース)で該当箇所だけを検索して渡すのは、この考え方の実装です。

MCP を使う理由も一言で。ツールごとに独自の接続を作ると、モデルやツールが増えるたびに組み合わせの数だけ実装が要ります。**共通の取り決めに合わせておけば、同じ約束事で新しいツールをつなげられる**というのが利点です。「MCP はモデルの精度を上げる技術」は誤りで、**つなぎ方の標準**です。

1-2 AIの実用的なユースケース [1.2]

1-2-1 AI/MLが価値を提供できる場面 [1.2]

AI/MLを使う理由は、突き詰めると三つです。設問では「この取り組みの主な価値はどれか」という形で問われます。

- **人の意思決定の支援**……人が最終判断を下す前に、根拠となる見込み・優先順位・要約を出す。例:与信担当者に申込ごとのリスクスコアを提示する、医師に読影の候補所見を提示する。**判断そのものを置き換えるのではなく、判断の材料を増やす**のがこの型です。
- **ソリューションのスケーラビリティ**……人手では処理量に上限がある作業を、**量が増えても同じ品質で捌ける**ようにする。例:1日1万件の問い合わせを自動で分類する、全商品SKUに対して毎日需要予測を出す。**人を増やさずに量に対応できる**ことが価値です。
- **自動化**……定型的で判断の余地が小さい作業を機械に任せ、人を高付加価値の作業に回す。例:請求書からの項目抽出、通話録音の文字起こしと要約。

もう一つ、AIならではの価値として**個別化(パーソナライズ)**があります。利用者ごとに違う推薦・違う案内を、人手では不可能な規模で出す。レコメンデーションシステムがその代表です。

取り違えの注意。「AIを導入すれば人員を削減できる」という選択肢は、AIF-C01 の文脈では**主たる価値の説明として弱い**です。試験が求める答えは、**意思決定の質を上げる・処理量に耐える・繰り返し作業を自動化する**という機能面の説明です。また「AIは常に人より正確」も誤りで、AIは**確率的な見込み**を返すため、誤りが起こる前提で運用を組みます。

1-2-2 AI/MLソリューションが適切でない場面 [1.2]

「AIを使わない」が正解になる設問が確実に出ます。判断の軸は次の四つです。

1. **決まった結果(決定的な出力)が必要な場面**……法令や社内規程で答えが一意に決まる処理、会計計算、税額の算出、給与の計算、閾値による単純な判定。これらは**ルールベースの仕組み**で書くべきで、確率的な予測を挟む理由がありません。「申込書の項目を条件表と突き合わせて可否を決める」は、AIではなく**ルールベースの自動化**が正解です。
2. **費用対効果が見合わない場面**……月に数十件しか発生しない作業のために、データ整備・学習・運用監視の体制を作る。**開発コストと運用コストが、削減できる工数を上回る**なら見送りが正解です。費用対効果分析は、**モデルの精度ではなく事業への効果で測る**点に注意します。
3. **説明できないと使えない場面**……なぜその結論になったのかを規制当局や顧客に説明する義務がある業務。詳しくは 1-2-6 で扱います。
4. **データが無い・偏っている・品質が低い場面**……学習の材料が揃っていなければ、どの手法を選んでも結果は出ません。「精度が出ないので別のアルゴリズムに替える」は、材料が原因なら的外れです。**欠損と重複を洗い出し、記録の取り方から整える**のが先です。

判別のための対比表。

課題	適する手段
消費税額を税率表に従って計算する	**ルールベース**(計算式が決まっている)
在庫が閾値を下回ったら自動発注する	**ルールベース**(条件が明確)
過去の実績から翌月の需要を見込む	**AI/ML**(正解が一意に決まらない)
契約書の中から特定の条項を探す	**AI/ML**(NLP。文章の意味を扱う)
社員番号から所属部署を引く	**データベース参照**(AIは不要)
年に数回しか起きない事象を予測する	**AI/MLは不向き**(事例が少なすぎる)

1-2-3 ユースケースに応じたAI/ML手法の選択 [1.2]

手法の選択は、**予測したい出力の形**で決まります。ここは組み合わせ問題の定番です。

手法	出力の形	学習の種類	業務例
回帰	**連続した数値**	教師あり	翌月の売上金額、機器の残り寿命、不動産の価格、来店客数
分類	**あらかじめ決めたカテゴリ**	教師あり	迷惑メールか否か(二値)、問い合わせの種別(多値)、良品/不良品、不正取引か否か
クラスタリング	**データをまとめたグループ**(名前は付かない)	教師なし	購買履歴からの顧客セグメント、似た故障パターンのまとめ

補助的な手法も名前と用途で。

- **異常検知**……通常の振る舞いから外れたものを見つける。ラベルが極端に偏る不正検知や設備故障の予兆で使います。
- **次元削減**……項目数が多すぎるデータを、情報を保ちつつ少ない軸にまとめる。可視化や前処理に使います。
- **レコメンデーション**……利用者と商品の関係から、次に好みそうなものを提示する。

判別の合言葉。

- 出力が**金額・件数・日数などの数値**なら**回帰**。
- 出力が**あらかじめ決めた選択肢のどれか**なら**分類**。
- **正解のラベルが無く、似たもの同士をまとめたい**なら**クラスタリング**。

よく出る取り違え。

場面	誤答	正解
顧客が解約するかどうかを見込む	回帰	**分類**(するかしないかの二値)
顧客が解約するまでの残り日数を見込む	分類	**回帰**(数値)
顧客をいくつかのタイプに分けたいが、タイプの定義は無い	分類	**クラスタリング**(教師なし)
問い合わせを既存の5カテゴリに振り分ける	クラスタリング	**分類**(カテゴリが決まっている)

「カテゴリが**あらかじめ決まっている**なら分類、**決まっておらずデータから見つける**ならクラスタリング」――この一行が最も効きます。

1-2-4 現実世界のAIアプリケーションの例 [1.2]

代表的な応用を、**課題の言い回しから分野を当てられる**ようにしておきます。

応用	中身	業務例
コンピュータビジョン	画像・動画の理解	製造ラインの外観検査、医用画像の所見候補、店舗の混雑把握、書類のスキャン読み取り
NLP(自然言語処理)	文章の理解・生成	問い合わせの自動分類、レビューの感情分析、契約書からの条項抽出、要約
音声認識	音声を文字に変換	コールセンターの通話の文字起こし、会議の議事録作成、音声での操作
レコメンデーションシステム	利用者に合う商品・記事を提示	EC の「あなたへのおすすめ」、動画配信の次に見る候補
不正検知	通常と異なる取引を見つける	クレジットカードの不正利用、保険金請求の異常、アカウントの乗っ取り
予測(フォーキャスティング)	将来の数値を見込む	需要予測、在庫の適正化、人員配置、電力需要
ナレッジベース	社内文書を根拠に質問へ答える	社内規程のQ&A、製品マニュアルの検索と回答、問い合わせ対応の下敷き
エージェント型AI	目標に対し自ら計画しツールを実行	予約の変更を関連システムまで含めて完了させる、調査から報告書作成までを複数手順で進める

ナレッジベースの位置づけは、v1.1 で明示された論点です。ナレッジベースは、**社内文書を検索して、その内容を根拠に生成AIに答えさせる**仕組み(RAG=検索拡張生成)を支える要素です。狙いは二つ ——**モデルが知らない社内固有の情報を答えられるようにすること**と、**根拠のある回答にしてハルシネーションを抑えること(グラウンディング)**。「ナレッジベースを使えばモデルを再学習したことになる」は誤りで、**モデル自体は変わらず、参照する材料を渡しているだけ**です。参照先の文書が古ければ古い答えが返るので、**参照先を最新の版だけに整理する**運用が要ります。

エージェント型AIとナレッジベースの違いも対で。ナレッジベースは**読んで答える**まで、エージェント型AIは**読んだうえで外部システムを操作して完了させる**ところまでを担います。

1-2-5 AWSのマネージドAI/MLサービスの能力 [1.2]

AIF-C01 では、**サービス名と用途の対応**が確実に問われます。まず「学習済みモデルをAPIで呼ぶだけで使えるAIサービス」の一覧です。

サービス	できること	典型的な出題の言い回し
Amazon Transcribe	**音声をテキストに変換**(自動音声認識)	「通話録音を文字起こししたい」「会議の音声から議事録を作りたい」
Amazon Translate	**テキストを別の言語に翻訳**	「商品説明を多言語で出したい」「海外拠点向けに文書を翻訳したい」
Amazon Comprehend	**テキストの中身を分析**。感情分析、キーフレーズ抽出、固有表現(エンティティ)抽出、言語判定、トピック分類。**PIIの検出**もできる	「レビューの評価が肯定的か否定的かを判定したい」「文書から個人情報を検出したい」

サービス	できること	典型的な出題の言い回し
Amazon Lex	**会話型インターフェース(チャットボット/音声ボット)**の構築。意図(インテント)とスロットで対話を設計	「対話で予約を受け付けるボットを作りたい」
Amazon Polly	**テキストを音声に変換**(音声合成)	「記事を読み上げさせたい」「音声案内を自動生成したい」
Amazon Rekognition	**画像・動画の分析**。物体やシーンの検出、顔の検出・比較、不適切コンテンツの検出、画像内の文字検出	「画像に何が写っているか判別したい」「不適切な投稿画像を弾きたい」
Amazon Textract	**文書から文字・表・フォーム項目を抽出**(OCRの発展形)	「請求書や申込書の項目を読み取って取り込みたい」
Amazon Personalize	**レコメンデーション**の仕組みを提供	「利用者ごとのおすすめを出したい」

Transcribe と Comprehend の違いは頻出です。**Transcribe は音声→テキストの変換まで、Comprehend はテキストの中身の分析**。通話の感情を知りたいなら、**Transcribe で文字にしてから Comprehend にかける**という二段構えが正解です。**Polly は逆向き(テキスト→音声)**で、Transcribe と混同させる選択肢が作られます。

Textract と Rekognition の違いも対で。**文書の項目を構造ごと取り出すなら Textract、写真に何が写っているかを見るなら Rekognition** です。Rekognition にも文字検出はありますが、**帳票のフォームや表の構造を取り出す用途は Textract** が適切です。

次に、モデルを自分で作る・基盤モデルを使うための土台となるサービスです。

サービス	位置づけ
Amazon SageMaker AI	**ML のライフサイクル全体**(データ準備、学習、チューニング、デプロイ、監視)をカバーするマネージドサービス。**自前のモデルを作る・運用する**ときの中心
Amazon SageMaker JumpStart	**事前学習済みモデルと構築済みソリューションのカタログ**。オープンソースのモデルを選んで、そのまま展開したりファインチューニングしたりできる
Amazon Bedrock	**基盤モデル(FM)をAPI経由で使う**ためのマネージドサービス。複数ベンダーのFMを共通のAPIで呼べる。ナレッジベース、エージェント、ガードレール、プロンプト管理などの機能を伴う
Amazon Nova	AWS が提供する**基盤モデルのファミリー**。Bedrock 経由で利用する
Amazon Q	**生成AIを使ったアシスタント**。業務の質問応答や作業の支援を行う
AWS Transform	既存のワークロードの**移行・近代化をAIで支援**するサービス

SageMaker AI と Bedrock の使い分けは、この試験の中心的な分岐点です。

要件	選ぶもの
自社データで**独自のモデルを一から学習**したい	**SageMaker AI**
学習・デプロイ・監視まで**MLのライフサイクル全体**を管理したい	**SageMaker AI**

要件	選ぶもの
既存の**基盤モデルをAPIで呼んで**要約・作文・チャットをさせたい	**Amazon Bedrock**
インフラを一切管理せずにFMを使いたい	**Amazon Bedrock**
公開されている**事前学習済みモデルを選んで展開**したい	**SageMaker JumpStart**

「Bedrock でモデルを一から学習する」「SageMaker AI はAPIを呼ぶだけのサービス」は、いずれも誤りとして出ます。

1-2-6 従来型MLと基盤モデル(FM)の使い分け [1.2]

v1.1 で新設された論点です。**「新しいからFM」ではなく、要件から選ぶ**という考え方が問われます。

観点	従来型ML	基盤モデル(FM)
得意な出力	数値の予測、分類、スコアリング	文章生成、要約、翻訳、対話、コード生成
必要なデータ	目的に合った**ラベル付きデータ**が要る	**事前学習済み**なので、少量の例やプロンプトで動く
説明可能性	**高い**(どの項目がどれだけ効いたかを示しやすい)	**低い**(なぜその文が出たかを説明しにくい)
出力の一貫性	**同じ入力に同じ出力**を返せる	**毎回変わりうる**。ハルシネーション(幻覚)が起こりうる
コスト構造	学習時に費用、推論は軽い	**トークン課金**で使うほど積み上がる。大きなモデルほど高価で遅い

観点	従来型ML	基盤モデル(FM)
開発期間	データ整備と学習に時間がかかる	短期間で試作できる
運用上の制約	小さいモデルなら**エッジや低スペック環境**でも動かせる	大きなモデルは**推論に相応の計算資源**が要る

選択の判断軸を3つ。

1. **規制上の懸念**……金融の与信判断、保険の引受、医療の判断のように、**説明責任・監査対応・再現性が法令や業界規制で求められる**場面では、**従来型ML**が適します。判断の根拠を項目単位で示せるためです。
2. **説明可能性の要件**……「なぜ否認されたのか」を顧客に説明する義務があるなら、**従来型ML**。文章生成の必要がなく、判断根拠の提示が要件なら、FMを持ち出す理由はありません。
3. **運用上の制約**……応答速度、推論コスト、オフライン環境、データを外に出せない制約。**リアルタイムで大量に安く回す必要があるなら従来型ML、多様な言語タスクを1つのモデルで賄いたいならFM**です。

従来型MLと基盤モデル(FM)の使い分け —— 要件から選ぶ

従来型ML	基盤モデル(FM)
得意な出力 数値の予測、分類、スコアリング	得意な出力 文章生成、要約、翻訳、対話
必要なデータ ラベル付きデータが要る	必要なデータ 事前学習済みなので、少量の例やプロンプトで動く
説明可能性 高い	説明可能性 低い
出力の一貫性 同じ入力に同じ出力	出力の一貫性 毎回変わりうる
コスト構造 学習時に費用、推論は軽い	コスト構造 トークン課金で使うほど積み上がる

選択の判断軸を3つ。

1. 規制上の懸念 —— 説明責任・監査対応・再現性が求められるなら従来型ML
2. 説明可能性の要件 ——「なぜ否認されたのか」を説明する義務があるなら従来型ML
3. 運用上の制約 —— リアルタイムで大量に安く回すなら従来型ML、多様な言語タスクならFM

「FMは従来型MLの上位互換なので常にFMを選ぶ」は誤り。
出力の形と、説明責任と、コストの三点で切り分ける。

図 1-4　従来型MLと基盤モデル(FM)の使い分け

判別の練習。

場面	選ぶべきもの	理由
与信審査で否認理由を顧客に説明する義務がある	**従来型ML**	説明可能性と再現性が要件
数百万件の取引を毎秒スコアリングする不正検知	**従来型ML**	推論コストと応答速度の制約
問い合わせメールへの返信文の下書きを作る	**FM**	生成が目的でラベル付きデータが無い
社内規程について自然文で質問に答えさせる	**FM+ナレッジベース**	生成が目的。根拠付けはグラウンディングで
需要予測の数値を毎日出す	**従来型ML**	出力が数値で、正解との誤差で評価できる

場面	選ぶべきもの	理由
ラベル付きデータが全く無く、まず試作して価値を確かめたい	**FM**	事前学習済みなので少ない準備で始められる

取り違えの注意。「FMは従来型MLの上位互換なので常にFMを選ぶ」は誤りです。**数値予測に FM を使うと、コストが高く、説明できず、出力が揺れる**という三重の不利を負います。逆に「従来型MLで文章を生成する」も無理があります。**出力の形と、説明責任と、コストの三点で切り分ける**のが正解の作法です。

1-2-7 場面別の判断演習(誤答肢の切り方) [1.2]

ドメイン1の設問は、場面を読んで「何の課題か」を言い当てられれば解けます。**誤りの選択肢がなぜ誤りなのか**まで言えるようにしておきます。

場面1:製造業の品質保証部が、外観検査の目視を減らしたい。正解は**コンピュータビジョンによる画像分類(教師あり学習)**です。良品・不良品のラベル付き画像を材料にします。誤答は、①生成AIで検査基準の文書を作らせる(文書を作ることと画像を判定することは別の作業)、②クラスタリングで画像をまとめる(良品か不良品かという既定のカテゴリがあるので分類が適切)、③ルールベースで閾値を書く(傷の見え方が多様で規則を書き下せない)です。

場面2:小売の販促担当が、顧客をタイプ別に分けて施策を変えたい。タイプの定義はまだ無い。正解は**クラスタリング(教師なし学習)**です。誤答は、①分類(振り分ける先のカテゴリが存在しない)、②回帰(出力が数値ではない)、③レコメンデーション(個人への商品提示であって、集団の分け方ではない)です。

場面3:コールセンターの通話から、不満を述べた顧客を抽出したい。正解は**Amazon Transcribe で文字にしてから Amazon Comprehend で感情分**

析という二段構えです。誤答は、①Amazon Comprehend だけを使う(入力が音声なのでそのままでは扱えない)、②Amazon Polly を使う(テキストから音声への逆向き)、③Amazon Translate を使う(翻訳であって感情の分析ではない)です。

場面4:経理部が、消費税額の計算を自動化したい。 正解は**AIを使わず、ルールベースで実装する**ことです。税率表から一意に決まるため、確率的な予測を挟む理由がありません。誤答は、①回帰で税額を予測する(決定的な結果が必要な場面でAIを使う典型的な誤り)、②FMに計算させる(出力が揺れうるうえ費用も高い)です。

場面5:与信審査に機械学習を使いたいが、否認理由を顧客へ説明する義務がある。 正解は**従来型MLを選び、説明可能性を確保する**ことです。誤答は、①LLMに審査させる(説明可能性が低く、規制上の要件を満たしにくい)、②説明は不要と整理する(規制上の懸念を無視している)です。

場面6:社内規程について、社員の自然文の質問に答えさせたい。規程は毎月更新される。 正解は**FM+ナレッジベース(RAG)**です。誤答は、①更新のたびにファインチューニングする(費用と期間が見合わず、更新の速さにも追いつかない)、②従来型MLで分類する(質問への文章での回答が求められている)、③モデルの再学習で対応する(参照先の差し替えで足りる)です。

場面7:月に数件しか発生しない特殊な事務処理を自動化したい。 正解は**AI/MLの導入を見送る**か、ルールベースで簡素に処理することです。**費用対効果**が成立せず、学習に足る事例も集まりません。誤答は、①少ないデータでも深層学習なら学習できる(むしろ大量のデータを要する)、②FMなら学習不要なので必ず適する(適するかは費用と要件次第で、常に正解ではない)です。

場面8:予約の変更を、在庫確認・決済・通知まで含めて完了させたい。 正解は**エージェント型AI**です。複数の外部システムを順に呼ぶ必要があるためです。

誤答は、①ナレッジベースで対応する(読んで答えるまでで、操作は行わない)、②単発のプロンプトで済ませる(手順の実行が伴わない)です。なお、この構成では**権限の最小化と操作の記録、重要な操作への人の承認**が併せて必要になります。

1-3 AI/ML開発ライフサイクル [1.3]

1-3-1 AI/MLパイプラインの構成要素 [1.3]

AI/MLの取り組みは、**決まった順番の工程(パイプライン)**として整理されています。順序付け問題と、「この作業はどの段階か」を問う設問の材料になります。

#	段階	やること	よく出る言い回し
1	**ビジネス課題の定義**	何の意思決定に使うのか、成功の基準は何かを決める	「解約率を何ポイント下げる」「対応時間を何分短縮する」
2	**ML課題への言い換え(フレーミング)**	回帰か、分類か、クラスタリングかへ翻訳する	「解約するかどうかを当てる二値分類にする」
3	**データ収集**	必要なデータをどこから取るかを決めて集める	「販売管理と問い合わせ履歴を統合する」
4	**データ前処理・準備**	欠損の処理、重複の排除、表記の統一、外れ値の扱い	「部門ごとに違う日付の書式をそろえる」

#	段階	やること	よく出る言い回し
5	**特徴量エンジニアリング**	予測に効く項目を作る・選ぶ	「直近3か月の購入回数を新しい列として作る」
6	**モデルの学習**	アルゴリズムを選び、学習データでモデルを作る	「学習用データでパラメータを調整する」
7	**モデルの評価**	**学習に使っていないデータ**で性能を測る	「検証データで適合率と再現率を確認する」
8	**チューニング**	ハイパーパラメータや特徴量を調整して改善する	「学習率や木の深さを変えて比較する」
9	**デプロイ(本番展開)**	推論できる形で本番に置く。バッチ・リアルタイム・非同期・サーバーレスから選ぶ	「エンドポイントを作って業務システムから呼ぶ」
10	**モニタリング(モデル監視)**	精度、データの傾向の変化、遅延、コストを見張る	「予測と実績の差を毎月集計する」
11	**再学習**	精度の劣化やデータの変化に応じて学習をやり直す	「直近データを足して四半期ごとに学習し直す」

データの分け方も、ここで押さえます。学習データ(トレーニング)・検証データ(バリデーション)・テストデータ(評価用)の三つに分け、**テストデータは最後の評価にだけ使います**。「学習に使ったデータで測った精度は95%です」という

報告を受けたら、求めるべきは**学習に使っていないデータで測り直した結果**です。同じデータで何度測っても、過学習を見抜けません。

生成AIのパイプラインは少し形が変わります。学習の代わりに**モデルの選定 → プロンプトの設計 → 必要なら RAG(ナレッジベース)の構築 → 必要なら ファインチューニング → 評価 → デプロイ → 監視**という流れになります。**多くの場合、自前の学習は行わず、既存のFMをどう使うかの設計が中心**である点が、従来型MLとの大きな違いです。

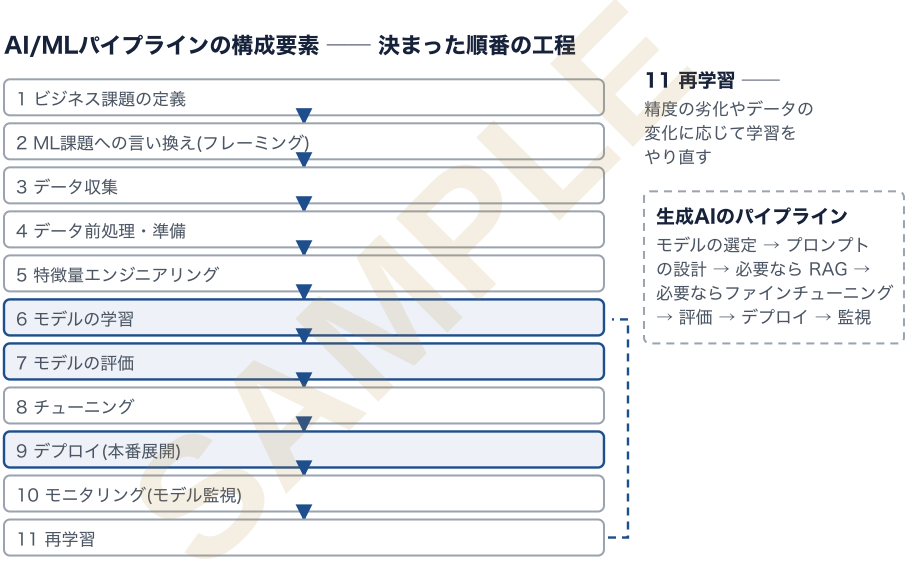


図 1-5　AI/MLパイプラインの構成要素 —— 決まった順番の工程

よく出る取り違え。

誤った説明	正しくは
特徴量エンジニアリングはデプロイの後の工程である	**学習の前**の工程

誤った説明	正しくは
モデルの評価は学習データで行う	**学習に使っていないデータ**で行う
デプロイが終われば工程は完了である	**監視と再学習**が続く
ビジネス課題の定義はモデルができてから行う	いちばん**最初**の工程

1-3-2 FMモデルの入手元 [1.3]

基盤モデルをどこから手に入れるかは、三つの選択肢に整理できます。

入手元	中身	向く場面	注意点
オープンソースの事前学習済みモデル	公開されている学習済みモデルをそのまま使う、または手元で調整する	コストを抑えたい、モデルを自社環境に置きたい、改変したい	**ライセンス条件の確認**が必須。運用と更新は自社の責任
商用の事前学習済みモデル(マネージド提供)	ベンダーが提供するFMをAPIで呼ぶ	すぐ使いたい、インフラを持ちたくない	使用量に応じた課金。モデルの中身は改変できない
カスタムモデルの学習	自社データでモデルを一から学習する、または既存モデルをファインチューニングする	独自ドメインの精度が要る、社内固有の表現が多い	**データ量・費用・期間が最大**。効果が見合うかの検討が要る

AWS上での対応は次のようになります。**オープンソースの事前学習済みモデルを探して展開するなら Amazon SageMaker JumpStart、商用FMをAPIで呼ぶなら Amazon Bedrock、一から学習する・本格的にファインチューニングするなら Amazon SageMaker AI**です。

適応の度合いは、軽い順に選ぶのが原則です。

1. **プロンプトエンジニアリング**……指示の書き方だけを工夫する。**コスト最小・即座に試せる**。
2. **RAG(検索拡張生成)/ナレッジベース**……社内文書を検索して根拠として渡す。**モデルは変えない**。最新情報や社内固有情報に対応でき、グラウンディングによりハルシネーションを抑えられる。
3. **ファインチューニング**……自社データで**モデルのパラメータを追加学習**する。文体・専門用語・出力形式を深く合わせたいとき。**データ準備と費用がかかる**。
4. **継続事前学習**……大量の自社ドメイン文書でさらに事前学習する。もっとも重い。

取り違えの注意。「社内の最新の規程に基づいて答えさせたい」に対する答えは、**ファインチューニングではなく RAG(ナレッジベース)**です。規程は更新されるため、そのたびに再学習するのは現実的でなく、**参照先の文書を差し替えるだけで済む RAG が適切**です。逆に「出力の文体やフォーマットを常に自社仕様に合わせたい」は、**ファインチューニング**が適します。

モデル蒸留もここで押さえます。**大きく高性能なモデル(教師)の振る舞いを、小さなモデル(生徒)に学習させて再現する**手法です。狙いは**推論コストと応答時間の削減**で、精度は多少落ちる代わりに、安く速く動くモデルが得られます。「精度を上げるための手法」ではない点に注意してください。

1-3-3 本番環境でモデルを使う方法 [1.3]

デプロイの形は、大きく二つです。

方法	中身	利点	欠点
マネージドAPIサービス	事業者が用意したエンドポイントをAPIで呼ぶ(Amazon Bedrock、Amazon Comprehend などのAIサービス)	インフラの管理が不要、すぐ使える、スケールが自動、運用の手間が小さい	モデルの中身を制御できない、使用量に応じた課金が積み上がる
セルフホスト型API	自分で用意した計算資源(Amazon EC2、Amazon ECS/Amazon EKS、SageMaker AI のエンドポイント)にモデルを載せて公開する	モデル・バージョン・配置場所を自分で制御できる、要件に合わせた最適化ができる	**インフラの構築・監視・拡張・更新が自社の責任**。待機コストがかかる

選択の判断軸。

- **早く始めたい・運用の人手が無い** → マネージドAPIサービス。
- **モデルを自社で持つ必要がある(規制・データの所在・独自モデル)** → セルフホスト型API。
- **要求が読めない・間欠的** → **サーバーレス推論**でホストして待機コストを避ける。
- **常時大量の要求がある** → **リアルタイム推論**のエンドポイントを常設。単価が下がる。
- **即時性が要らない大量処理** → **バッチ推論**。

共有責任の観点も一言だけ。マネージドサービスを使えばインフラの運用はAWS側が担いますが、**どのデータを入力するか、出力をどう検証するか、誰にアクセスを許すか**は常に利用者側の責任です。「マネージドAPIにすれば出力の妥当性の確認も不要になる」は誤りです。

1-3-4 AI/MLパイプラインの各段階に対応するAWSサービス [1.3]

v1.1 でこの項目は「Amazon Bedrock、Amazon Quick、Kiro、SageMaker AI」を例として挙げる形に変わりました。**各サービスがパイプラインのどの段階に当たるか**を対応付けて覚えます。

段階	主なサービス	役割
データの蓄積	**Amazon S3**、Amazon Redshift、Amazon RDS/Amazon Aurora、Amazon DynamoDB	学習データ・文書・成果物の保管先。S3 がデータレイクの中心
データの準備・変換	**AWS Glue**、AWS Glue DataBrew、Amazon EMR、AWS Lake Formation、AWS Data Exchange	抽出・変換・結合、品質の整備、権限の一元管理、外部データの調達
可視化・分析	**Amazon Quick**	データを可視化し、**自然言語での問いかけからダッシュボードや分析結果を得る**。パイプラインの入口(現状把握)と出口(結果の共有・意思決定)の両方で使う
モデルの構築・学習・チューニング	**Amazon SageMaker AI**、Amazon SageMaker JumpStart	ノートブック環境、学習ジョブ、チューニング、実験管理。JumpStart は事前学習済みモデルのカタログ
基盤モデルの利用・カスタマイズ	**Amazon Bedrock**	FMをAPIで呼ぶ。ナレッジベース(RAG)、エージェント、ガードレール、**Prompt Management**(プロンプトのバージョン管理)、評価機能を伴う

段階	主なサービス	役割
エージェントの構築・運用	**Amazon Bedrock AgentCore**、**Strands Agents**	AgentCore はエージェントの実行基盤(Identity で認証・認可、Policy で行動の制約)。Strands Agents は構築用のオープンソース SDK
アプリケーション開発	**Kiro**	AIを前提とした開発環境(IDE)。仕様の整理から実装までを支援し、**AI機能を組み込むアプリを作る側**の道具
デプロイ・実行基盤	SageMaker AI のエンドポイント、**AWS Lambda**、Amazon ECS/Amazon EKS、Amazon EC2	推論の実行場所。Lambda はサーバーレスの処理、ECS/EKS はコンテナ、EC2 は自前構成
監視・記録	**Amazon CloudWatch**、**AWS CloudTrail**、AWS Config	指標とログの監視、API呼び出しの記録(監査ログ)、構成の追跡
セキュリティ	**IAM**、AWS KMS、Amazon Macie、AWS Secrets Manager、Amazon Inspector	権限の付与、暗号化の鍵管理、機微データの検出、資格情報の保管
コスト管理	**AWS Cost Explorer**、AWS Budgets	利用費用の把握と予算の超過検知

この対応表で押さえる4つの主役。

- **Amazon Bedrock……FMを使う段階**。作らずに借りる側。
- **Amazon SageMaker AI……モデルを作る・運用する段階**。ライフサイクル全体。

- **Amazon Quick……データを見る・結果を共有する段階。**可視化と自然言語での分析。
- **Kiro……アプリを作る段階。**開発者の作業環境。

よく出る取り違え。

誤った対応付け	正しくは
Kiro でモデルを学習する	Kiro は**開発環境。**学習は SageMaker AI
Amazon Quick で基盤モデルを呼び出して業務アプリを作る	Quick は**可視化・分析。**FM利用は Bedrock
Bedrock でハイパーパラメータチューニングを行う	それは **SageMaker AI** の領域
SageMaker AI は事前学習済みFMを呼ぶ専用サービス	SageMaker AI は**構築から運用まで。**FMのAPI利用は Bedrock、事前学習済みモデルのカタログは JumpStart
CloudTrail でモデルの精度を監視する	CloudTrail は**API呼び出しの記録。**精度や指標の監視は CloudWatch とモデル監視の仕組み

1-3-5 MLOps(MLオペレーション)の基本概念 [1.3]

MLOps は、**モデルを一度作って終わりにせず、繰り返し安全に改善し続けられる状態を作る取り組み**です。試験では、用語の意味と「その課題はどの概念に当たるか」が問われます。

概念	中身	対応する困りごと
実験(エクスペリメンテーション)	複数の条件を試し、**どの設定でどの結果が出たかを記録して比較できる**ようにする	「誰がどのデータでどの設定を試したか分からない」

概念	中身	対応する困りごと
再現可能なプロセス	データ・コード・設定のバージョンを管理し、**同じ手順で同じモデルを作り直せる**ようにする	「半年前のモデルを作り直せない」
スケーラブルなシステム	データ量や要求数が増えても**性能と運用が破綻しない**設計にする	「対象店舗を10倍にしたら学習が終わらない」
技術的負債の管理	場当たりの処理、手作業の手順、放置されたモデルなど、**後で高くつく作りを溜め込まない**	「本番に誰も内容を知らないモデルが5個ある」
本番運用準備(プロダクションレディネス)	監視、権限、障害時の切り戻し、性能要件、文書化を整えてから本番に出す	「実験環境のまま本番に出して障害時に戻せない」
モデル監視(モニタリング)	精度、入力データの傾向、応答時間、コストを継続的に見張る	「精度が落ちていたことに半年気づかなかった」
モデル再学習(リトレーニング)	劣化や変化に応じて学習をやり直し、検証してから差し替える	「市場の傾向が変わって予測が外れ始めた」

本番へ出すまでの順序も問われます。先に**再現可能なプロセス**（データ・コード・設定のバージョン管理）を整え、次に**本番運用準備**（監視・権限・切り戻し・性能要件・文書化）を済ませ、そのうえで**一部の利用者にだけ新しいモデルを出して旧モデルと比較**し、問題が無いことを確かめてから**全体へ展開して継続的に監視**します。バージョン管理より先に本番運用準備に入ると、問題が起きた時にどの版へ戻せばよいかが分からなくなります。

モデルの劣化(ドリフト)は、監視と再学習の理由そのものなので、言葉を分けて覚えます。

- **データドリフト……入力データの傾向が学習時と変わる**こと。例:新しい年齢層の顧客が増えた、新商品が追加された。
- **コンセプトドリフト……入力と正解の関係そのものが変わる**こと。例:感染症や制度変更で購買行動の意味が変わった。

どちらも「モデルは変わっていないのに精度が落ちる」現象を生みます。対応は、**監視で早く気づき、直近のデータを含めて再学習し、評価してから差し替える**という順序です。「精度が落ちたので即座に本番のモデルを差し替える」は、評価を飛ばしている点で誤りです。

MLOps の運用でよく問われる原則。

- **自動化する対象は、データの取り込みから学習・評価・デプロイまでの一連の手順**です。手作業の手順書は再現性が担保できません。
- **モデルはバージョンで管理し、どのデータ・どのコードから作られたかを追跡できるようにします。**監査や不具合の追跡に必要です。
- **本番への展開は段階的に**行います。一部の利用者にだけ新しいモデルを出して比較する、旧モデルへ即座に戻せるようにする、といった手当てです。
- **監視の対象は精度だけではありません。**応答時間、エラー率、推論コスト、入力データの分布の変化、そして生成AIでは出力の安全性も含みます。

よく出る取り違え。

誤った説明	正しくは
MLOps はモデルの精度を上げるための手法である	MLOps は**運用の仕組み**。精度向上は結果として得られることがあるが目的ではない
デプロイの自動化ができていれば MLOps は完成である	監視・再学習・再現性まで含めて MLOps
再学習は定期的に必ず行うべきである	**必要性は監視の結果から判断する**。無意味な再学習はコストと不安定さを招く
実験の記録は開発者個人のメモで足りる	比較と再現ができる形で**共有される記録**が要る

1-3-6 モデル性能指標とビジネス指標 [1.3]

評価には**二つの物差し**があり、両方を使い分けることが問われます。

モデル性能指標(技術的な物差し)

分類の指標は、**混同行列**(実際と予測の組み合わせ)から導かれます。用語を先に固めます。

- **真陽性(TP)**……実際に陽性で、陽性と予測した。
- **偽陽性(FP)**……実際は陰性なのに、陽性と予測した(空振り)。
- **偽陰性(FN)**……実際は陽性なのに、陰性と予測した(見逃し)。
- **真陰性(TN)**……実際に陰性で、陰性と予測した。

指標	意味	一言で	重視すべき場面
正解率(Accuracy)	全体のうち正しく当てた割合	「全体の正しさ」	各クラスの件数が偏っていないとき

指標	意味	一言で	重視すべき場面
適合率(Precision)	陽性と予測したもののうち、実際に陽性だった割合	**「空振りの少なさ」**	誤検知の代償が大きいとき(正常な取引を止めてしまうと顧客が困る)
再現率(Recall)	実際に陽性のもののうち、陽性と当てられた割合	**「見逃しの少なさ」**	見逃しの代償が大きいとき(病気の見落とし、重大な不正の見逃し)
F1スコア	適合率と再現率の調和平均	**「両者のバランス」**	適合率と再現率のどちらも落としたくないとき
AUC(ROC曲線下面積)	閾値を変えたときの分離性能	「閾値によらない識別力」	陽性が少ないデータでの比較

回帰の指標も名前だけ押さえます。**MAE(平均絶対誤差)**は誤差の平均、**RMSE(二乗平均平方根誤差)**は大きな外れを重く見る誤差、**R二乗**は当てはまりの良さです。回帰に適合率や再現率は使えません(逆も同じ)。

正解率の落とし穴は最頻出です。不正取引が全体の0.1%しかないデータでは、**すべて「不正でない」と答えるだけで正解率は99.9%**になります。それでも不正は一件も見つけられていません。**クラスが偏っているときは、正解率ではなく適合率・再現率・F1スコア・AUCで見る**のが正解です。

適合率と再現率のトレードオフも対で覚えます。判定の閾値を下げれば、拾う件数が増えて**再現率は上がるが適合率は下がる**。上げれば逆になります。どちらを優先するかは、**見逃しの損失と空振りの損失のどちらが大きいか**という業務判断で決めます。

- 医療の重篤な疾患のスクリーニング → **再現率**を優先(見逃したくない)。

- 迷惑メールの自動削除 → **適合率**を優先(正常なメールを消したくない)。
- 不正検知で、疑わしいものは人が確認する運用 → **再現率**寄りにして、確認の工数と釣り合う点で閾値を決める。

生成AIの評価は、正解が一意でないため物差しが変わります。人による評価、**LLM-as-a-judge**(別のモデルに出力の良し悪しを判定させる方法。人の評価より安価で速いが、判定側の偏りが乗る)、既定の評価用データセットによる自動採点などを組み合わせます。エージェント型AIでは、**タスク完了率**(与えられた仕事を最後までやり遂げた割合)、**利用者の満足度**、**対話あたりコスト**といった指標が使われます。「生成AIの品質も正解率で測る」は誤りです。

ビジネス指標(事業の物差し)

技術的に良いモデルが、事業に効いているとは限りません。だから並行して測ります。

指標	意味	使いどころ
ユーザーあたりコスト	利用者1人(または1リクエスト・1対話)あたりにかかる費用	生成AIはトークン課金で積み上がるため、規模拡大の可否判断に直結する
開発コスト	データ整備、学習、実装にかかった費用と工数	費用対効果分析の分母
顧客フィードバック	満足度、評価、苦情の件数、利用継続率	出力の品質を事業側から測る
投資対効果(ROI)	得られた効果を投資額で割った比率	継続・拡大・中止の判断
業務のKPI	対応時間、解決率、解約率、在庫回転、売上	AI導入の狙いそのものを測る

両方を並べて判断するのが試験の求める姿勢です。

状況	正しい読み方
モデルの正解率は高いが、業務の対応時間が短縮していない	**業務に組み込まれていない**か、測る対象がずれている。ビジネス指標で見直す
精度は十分だが、推論コストが削減効果を上回っている	ROI が成立していない。**小さいモデルへの蒸留、バッチ推論への変更、プロンプトの短縮**などで単価を下げる
精度は少し低いが、対応時間が半減し満足度も上がった	事業としては成功。**モデル性能指標だけで打ち切らない**
実験環境の精度は高いが、本番で落ちた	**過学習**か**データドリフト**。学習に使っていないデータでの評価と監視に戻る

AIの取り組みをビジネス目標に整合させるという考え方も、v1.1 で明示されました。着手の順序は常に、**ビジネス目標 → 測る指標 → 手法の選択**です。「まず高精度のモデルを作り、使い道は後で考える」は順序が逆で、費用対効果の説明ができなくなります。

1-3-7 段階・成果物・関与するサービスの総ざらい [1.3]

順序付け問題と組み合わせ問題に備えて、**段階ごとの成果物**まで対応付けておきます。成果物が言えれば、工程の順序も自然に決まります。

段階	その段階の成果物	主に関与するもの
ビジネス課題の定義	目標と成功の基準(ビジネス指標)	事業部門
ML課題への言い換え	回帰/分類/クラスタリングのいずれかという方針	事業部門とデータ担当

段階	その段階の成果物	主に関与するもの
データ収集	学習に使えるデータの集まり	Amazon S3、AWS Data Exchange
データ前処理・準備	欠損と重複を整理し表記をそろえたデータ	AWS Glue、AWS Glue DataBrew、Amazon EMR
特徴量エンジニアリング	予測に効く項目を含む学習用データセット	Amazon SageMaker AI
学習	学習済みモデル	Amazon SageMaker AI
評価	正解率・適合率・再現率・F1スコアなどの評価結果	Amazon SageMaker AI
デプロイ	推論できるエンドポイントまたはバッチ処理の仕組み	SageMaker AI、AWS Lambda、Amazon ECS/Amazon EKS
モデル監視	精度とデータの傾向の推移の記録	Amazon CloudWatch
再学習	更新されたモデルと、差し替えの判断材料	Amazon SageMaker AI

生成AIの場合の対応も並べておきます。

段階	成果物	主に関与するもの
ユースケースの定義	何を生成させるか、成功の基準	事業部門
モデルの選定	用途・費用・応答速度に合うFMの決定	Amazon Bedrock、Amazon SageMaker JumpStart
プロンプトの設計	指示文とその版の管理	Amazon Bedrock Prompt Management

段階	成果物	主に関与するもの
根拠付け(グラウンディング)	参照する社内文書のナレッジベース	Amazon Bedrock のナレッジベース
カスタマイズ(必要なら)	ファインチューニング済みモデル、または蒸留した小型モデル	Amazon Bedrock、Amazon SageMaker AI
評価	人による評価、LLM-as-a-judge、タスク完了率、満足度、対話あたりコスト	Amazon Bedrock の評価機能
デプロイ	アプリケーションへの組み込み	Kiro(開発)、AWS Lambda ほか
エージェント化(必要なら)	ツール接続・権限・行動の制約を備えたエージェント	Amazon Bedrock AgentCore、Strands Agents、MCP
監視	出力の品質、コスト、監査ログ	Amazon CloudWatch、AWS CloudTrail

押さえどころは二つ。①生成AIでは「学習」の段階が無いことが多い(既存のFMを使うため)。②**評価の物差しが変わる**(正解率ではなく、人の評価やタスク完了率など)。この二点を混同させる選択肢がよく作られます。

1-4 この章のまとめ [1.1/1.2/1.3]

最後に、設問で迷ったときに戻る一行を並べます。

- **AI ⊃ ML ⊃ 深層学習**。ニューラルネットワークは構造の名前。GenAI は深層学習の応用、エージェント型AI は GenAI の応用。

- **学習は内部の値が変わる工程、推論は変わらない工程。**賢くするには再学習。
- **推論の形は4つ。**即時ならリアルタイム、大量一括ならバッチ、大きい入力や長時間なら非同期、間欠ならサーバーレス。
- **答えがあるなら教師あり学習、まとめたいなら教師なし学習、試行錯誤なら強化学習。**
- **出力が数値なら回帰、決まったカテゴリなら分類、カテゴリが無いならクラスタリング。**
- **一意に決まる処理はAIではなくルールベース。**費用対効果が合わないならやらないのが正解。
- **説明責任・再現性・低コストの大量推論なら従来型ML、生成が目的で準備を軽くしたいならFM。**
- **音声→テキストは Amazon Transcribe、テキスト分析は Amazon Comprehend、翻訳は Amazon Translate、対話は Amazon Lex、テキスト→音声は Amazon Polly。**
- **モデルを作るなら Amazon SageMaker AI、FMを借りるなら Amazon Bedrock、可視化と分析なら Amazon Quick、開発環境は Kiro。**
- **偏ったデータでは正解率は無意味。**見逃したくないなら再現率、空振りしたくないなら適合率、両立なら F1スコア。
- **モデル性能指標とビジネス指標(ユーザーあたりコスト、開発コスト、顧客フィードバック、ROI)は必ず両方で見る。**

サンプルはここまでです

続き(第2章〜巻末)は、AWS AIF-C01 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。

AWS AIF-C01 学習用テキスト(無料サンプル)

版

2026年7月版(AWS 公式 Exam Guide（AIF-C01・バージョン 1.1・2026年4月30日）準拠)

発行

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

お問い合わせ

info@shikaku-doujou.com