IT資格道場

Microsoft 認定 AI 変革リーダー(AB-731) 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)

AB-731 出題範囲(Skills measured as of January 11, 2027)準拠
2026年11月版

SAMPLE

ご利用にあたって

- このテキストは「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。
- 無料サンプルも著作物です。転載・改変しての再配布・二次販売はできません。
- AI モデルの学習データとしてのご利用(学習目的の入力・データセット化を含む)はできません。
- 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。

SAMPLE

目次

この試験の範囲 [overview]

第1章 Identify the business value of generative AI solutions [1.1.1/1.1.2/1.1.3/1.1.4/1.1.5/1.1.6/1.2.1/1.2.2/1.2.3/1.2.4/1.2.5/1.2.6/1.2.7/1.2.8/1.2.9]

- 生成 AI と他の種類の AI [1.1.1]
- 業務の要件から生成 AI の解決策を選ぶ [1.1.2]
- 事前学習済みモデルと微調整 [1.1.3]
- トークンと費用と ROI [1.1.4]
- 生成 AI の課題と緩和策 [1.1.5]
- 生成 AI が価値を出す場面 [1.1.6]
- プロンプト エンジニアリングが出力に与える影響 [1.2.1]
- プロンプト エンジニアリングの手法 [1.2.2]
- グラウンディングの要件 [1.2.3]
- RAG の使い方 [1.2.4]
- データが AI に与える影響 [1.2.5]
- 安全な AI の重要性 [1.2.6]
- 機械学習が価値を出す場面 [1.2.7]
- 機械学習のライフサイクル [1.2.8]
- AI システムのセキュリティ [1.2.9]
- 取り違えやすい組み合わせ [1.1.3/1.1.4/1.2.1/1.2.3/1.2.6/1.2.8/1.2.9]

サンプルはここまでです

SAMPLE

3 冊の使い方は、ダウンロード欄の案内を参照してください。

SAMPLE

この試験の範囲 [overview]

AB-731(Microsoft Certified: AI Transformation Leader)は、入門の試験です。生成 AI のビジネス価値、Microsoft の AI アプリとサービス、AI を組織に入れる戦略を問います。コードを書く力は問いません。

- レベル：Beginner
- 試験時間：45 分
- 合格：1000 点満点の 700 点以上
- 言語：日本語で受験できます

出題範囲は 3 つの分野です。この教材も同じ順に章を立てます(下の表)。配点は、学習ガイドの「Skills at a glance」にある割合です。

分野	配点	内容
1 Identify the business value of generative AI solutions	35〜40%	生成 AI の選び方・費用・課題、プロンプト・グラウンディング・RAG・データ、安全と機械学習
2 Identify benefits, capabilities, and opportunities for Microsoft's AI apps and services	35〜40%	Microsoft Copilot とその周辺、Copilot Studio、Foundry Tools と Microsoft Foundry
3 Identify an implementation and adoption strategy for Microsoft's AI apps and services	20〜25%	責任ある AI とガバナンス、導入チームと障壁、チャンピオン、ライセンスと課金

SAMPLE

学習ガイドの細目は 39 個。節の見出しの後ろにある [1.1.1] のような印が、細目の番号。

版の違いは次のとおりです。

- この教材は 2027 年 1 月 11 日版(Skills measured as of January 11, 2027)の学習ガイドに合わせています。
- 7 月 22 日版との違いは、製品名が Microsoft 365 Copilot から Microsoft Copilot に替わったことと、細目 3.2.5 のライセンスの箇条の書き替えです。分野と配点は同じです。

製品名は 2026 年 11 月時点の呼び名。旧名は併記します。

SAMPLE

第1章 Identify the business value of generative AI solutions [1.1.1/1.1.2/1.1.3/1.1.4/1.1.5/1.1.6/1.2.1/1.2.2/1.2.3/1.2.4/1.2.5/1.2.6/1.2.7/1.2.8/1.2.9]

この章の主題は 3 つです。生成 AI が何者で、どう選び、いくらかかるか。どんな弱点があり、どんな場面で価値を出すか。プロンプト・グラウンディング・RAG・データ・セキュリティ・機械学習をどう使い分けるか。

配点は 35〜40% で、3 分野で最大です。前半の 1.1 が基本の概念と選び方、後半の 1.2 が生成 AI の利点と機能、安全、機械学習です。

製品名は次のように読みます。学習ガイドの Microsoft Copilot は、以前の Microsoft 365 Copilot と同じ製品です。問題文に旧名と新名のどちらが出ても、別の製品とは読みません。[1]

生成 AI と他の種類の AI [1.1.1]

3 種類の AI

AI は、目的の違いで 3 種類に分けて考えます。

種類	何をするか	典型
従来型 AI（非生成・機械学習）	定められた目的のためにデータを分析して結果を出す	予測・分類・異常検知
生成 AI	ユーザーの依頼と利用できる文脈から、新しいコンテンツや応答を作る	文書作成・質問応答・要約・コード生成・従業員支援

SAMPLE

種類	何をするか	典型
エージェント型 AI	指示とツールを持つ AI モデルで、目標を追ってタスクを完了させる	依頼の解釈、判断、権限内での連携システムの利用

3 種類は組み合わせられます。たとえばエージェントが、生成 AI で依頼を解釈し、従来型 AI で需要を予測してから、権限の範囲内で行動します。AI 導入の論点は「使うか否か」ではなく「どこで価値を出すか」です。[2]

間違えやすいのは、生成 AI が万能の後継だという見方です。表形式の構造化データに対する時系列予測・回帰・分類は、古典的な機械学習の最も一般的な用途で、今も従来型 AI の領分です。[3]

アプリが必要とする出力は、判別モデル（データを分けて判定する）、生成モデル、またはその組み合わせです。[4]

生成 AI と LLM の仕組み

生成 AI は、自然言語・画像・音声・コードなどの独自コンテンツを作れる AI の一種です。学習データのパターンを使い、統計的に似た新しいコンテンツを作ります。毎回異なる内容を作るよう設計されているので、同じ入力でも同じ出力にはなりません。[5]

大規模言語モデル（LLM）は、大量のテキストで学習し、並びの中の次のトークンを予測するニューラルネットワークです。言語の理解と生成に関わるタスクが得意で、推論・文章生成・要約・翻訳・コード生成・分類や抽出・マルチモーダル理解が挙がります。LLM 単体は、テキストを入れてテキストを返す限られたシステムです。エージェントは、そこにツール・記憶・行動を足したものです。[6]

モデルは段階を重ねて作られます。事前学習でモデルは知識の大半を学び、事後学習でベースモデルが役立つアシスタントに変わります。さらに強化学習

SAMPLE

で段階的に考えることを学ばせたものが、推論モデルです。

モデルの種類	ふるまい
ベースモデル	続きを補完する
SFT と RLHF を経たモデル	指示に従う
推論モデル	段階的に考える

推論モデルは、数学・論理・コーディング・複数手順の問題に強い一方で、遅延とトークン使用量が増えます。精度が上がっても、遅延と費用は下がりません。

判別モデルと生成モデルの違いは、出力の種類です。判別モデルは、新しいコンテンツを作らず、予測や分類のような判定の結果を返します。生成モデルは、新しいコンテンツを作ります。顧客の解約を予測する仕事は判別（分類）の領分で、解約を防ぐ案内の文面を起こす仕事は生成の領分です。両方が要る業務は、片方だけでは成り立ちません。

場面で考えます。コールセンターが、通話の履歴から来週の入電数を予測し、問い合わせへの返信の下書きも作りたいとします。入電数の予測は、従来型 AI の時系列予測です。返信の下書きは、予測では作れません。生成 AI の仕事です。この 2 つを 1 つの仕組みに入れ、権限の範囲内で返信の送信まで進めるなら、エージェント型 AI の領分になります。

業務の要件から生成 AI の解決策を選ぶ [1.1.2]

選ぶ手順

最初に決めるのは、製品の名前ではありません。出発点は AI のユースケースで、Microsoft の AI decision tree は、その目的を 2 つに分けます。

SAMPLE

- 個人の効率向上: 起草・調査・会議準備・情報発見などで従業員の仕事を助ける
- 業務機能の自動化: エージェント・予測・意思決定支援などで業務を拡張・自動化する

原則は 1 つです。要件を満たす解決策のうち、運用責任が最も少ないものを承認します。カスタマイズが大きいほど、専門スキルも運用の持ち主としての責任も大きくなります。だから既製のものから先に検討します。[7]

目的別の解決策と費用の出どころ

目的・条件	選ぶもの	主な費用の出どころ
組織全体の個人効率（仕事と Web の文脈）	Microsoft Copilot	ライセンス
主に Web データのチャットで足り、フル体験は不要	Copilot Chat	無料
Azure・GitHub・Fabric・Dynamics 365・Power Platform などの製品内	製品内の Copilot	基本製品のサブスクリプション、または含まれる
セキュリティ・営業・サービス・財務などの職務	ロール別 Copilot	ライセンス、または含まれる
Microsoft Copilot を専門知識とエージェントで拡張	拡張ツール	Microsoft Copilot のライセンスと従量課金
ローコードや自然言語でエージェントを作る	Microsoft Copilot Studio	エージェントの利用に応じたクレジット
カスタムなエージェントやアプリを作る	Microsoft Foundry	モデルの利用とその他のツールの利用

SAMPLE

目的・条件	選ぶもの	主な費用の出どころ
自社データで ML モデルを学習・デプロイ	Azure Machine Learning	計算資源と周辺（ストレージ・データ転送）
すでに Fabric でデータ分析をしている組織の ML	Microsoft Fabric	容量・ストレージ・レプリケーション

決定木は、上の表に載せなかった行き先も 3 つ挙げます。

- 既製の非生成モデルや、エージェントを支える Azure AI Search が要る: Foundry Tools
- GPU の仮想マシンを管理せずに AI 推論や小さなモデルを動かす: Azure Container Apps（serverless GPU）。費用は計算資源と、モデルの利用・ストレージ・データ転送などの周辺
- 自前のモデルやランタイムを持ち込む: Azure の仮想マシンか Azure Kubernetes Service。費用は計算資源と、モデルの利用・ストレージ・データ転送などの周辺

Copilot Studio は、エージェントを作って組織のデータやシステムにつなぐ、マネージドなプラットフォームです。Foundry は、カスタムなエージェントやアプリを作るマネージドな Azure のプラットフォームです。

Foundry や Azure のプラットフォーム（PaaS）に進むのは、SaaS ではモデル・アプリのロジック・データ統合・AI 開発の制御が足りない時です。Azure のインフラ（自前のモデルとランタイム）は、AI 導入の既定の方法にしません。[8]

複数のプラットフォームにまたがる、対応する AI エージェントの可視化とガバナンスは、Microsoft Agent 365 が中央で担います。[9]

場面で考えます。営業部門が、社内の見積もりの手順に沿って問い合わせに答え、承認依頼まで出すエージェントを、開発者なしで半月以内に作りたいと

SAMPLE

します。目的は業務機能の自動化で、必要なのはローコードの作成です。運用責任が最も少ないのは Copilot Studio で、費用はエージェントの利用に応じたクレジットです。モデルや検索の制御まで握る必要が出るまでは、Foundry は要りません。

Copilot Studio・Foundry・拡張ツールの違いは、作り方と持ち主の責任の重さです。拡張ツールは、すでに使っている Microsoft Copilot に専門知識とエージェントを足します。Copilot Studio は、ローコードと自然言語で作る SaaS で、プラットフォームの運用を自分で持つ必要はありません。Foundry は、Azure の PaaS で、モデルの選択・アプリのロジック・データ統合まで細かく制御できる代わりに、専門スキルと運用の責任が増えます。

次の 3 つの場面で、同じ手順を当てはめます。

- 全社員 3,000 人の資料作成と会議の準備を助けたい。専門の開発者はいない。目的は個人の効率向上で、対象は組織全体なので、Microsoft Copilot です。費用はライセンスです。Web 上の情報への質問が中心の部署なら、無料の Copilot Chat で足りるかを先に確かめます。
- 営業支援の Dynamics 365 の画面の中で、商談の要約を作りたい。製品内の Copilot が最短です。別の基盤を作る必要はありません。
- すでに Microsoft Copilot を使っていて、人事の専門知識を持つエージェントを足したい。拡張ツールが合い、費用は Microsoft Copilot のライセンスと従量課金です。

自社専用のモデルを GPU の仮想マシンで動かす案が出ても、既定にはしません。SaaS と PaaS で要件を満たせないと確かめるまでは、Azure のインフラを検討しません。

SAMPLE

モデルの販売元で選ぶ

Foundry のモデルは、Azure が販売するモデルと、パートナーやコミュニティのモデルに分かれます。後者には、Hugging Face のハブにあるようなオープンなモデルも含まれ、作成と保守は提供元が持ちます。Azure 販売モデルは、Azure との統合・サポート・エンタープライズの SLA が要る場面向きです。パートナーやコミュニティのモデルは、専門的な用途や革新に向き、サポートはモデルの提供元が担います。[10]

場面で考えます。金融機関が、Azure との統合、サポート、企業向けの SLA を条件にするなら、Azure 販売モデルです。ただし、審査済みであっても、自社の文脈での評価は省けません。特定の専門分野に強い新しいモデルを試すなら、パートナーやコミュニティのモデルが候補になり、サポートの窓口は提供元です。

事前学習済みモデルと微調整 [1.1.3]

ファインチューニング

ファインチューニングは、事前学習済みのモデルに、特定のタスクやデータで追加の学習をして、性能向上・新スキルの追加・精度向上を図る手法です。医療・金融・法律などの専門分野への適応が、代表的な用途です。新しい知識を足す手段としては選びません。用語や専門知識の理解のほか、文体とトーン、指示への追従、言語や文化への適応、コンプライアンスも調整できます。[11]

手法は 3 つあります。

手法	学び方	使い所
SFT（教師ありファインチューニング）	入力と出力のペアで学習する	基本の手法。多くの用途に対応し、多くの顧客はここから始める

SAMPLE

手法	学び方	使い所
DPO（直接選好最適化）	好ましい応答と好ましくない応答の比較から学ぶ。別の報酬モデルは要らない	応答の質と安全性を高め、人の好み（好ましい文体やトーンなど）に寄せたい時
RFT（強化ファインチューニング）	報酬信号に基づく強化学習で最適化する	より複雑な最適化の目標。効果的に実装するには、MLの専門知識がより多く要る

データ量の目安は、初期テストが高品質な例 50〜100 件、本番用のモデルは 500 件以上です。実行の形は、多くの顧客には、使いやすさ、コスト効率、高性能モデルへのアクセスの面で、サーバーレスが最適とされます。分類・感情分析・コンテンツモデレーションなど絞った用途や、知識蒸留には、GPT-4.1-mini から始めると速く低コストです。

ファインチューニングと RAG

道具は、手元の便利さでは選びません。目的で選びます。ふるまい・文体・タスク性能を変えたい時が、ファインチューニングの出番です。新しい知識を足す手段ではありません。社内データや頻繁に変わる情報に基づく回答が要る時は、RAG を使います。[12]

場面で考えます。保険会社が、毎月改定される約款の内容に正確に答えるチャットを作ります。約款は変わり続けるので、学習し直す方法では追いつきません。取得して根拠にする RAG が正解です。ファインチューニングで知識を足す案は選びません。同じ会社が、返信の口調を自社の文体に揃えたい時は、ファインチューニングが向きます。

手段の選び分けを 1 枚にまとめます。

SAMPLE

変えたいもの	手段
その場の指示・出力の形（一度きり）	プロンプト（フューショットを含む）
新しい知識・頻繁に変わる社内の情報	RAG
ふるまい・文体・タスク性能・専門分野への適応	ファインチューニング

専門の用語と文体を身につけさせ、しかも最新の指針に沿って答えさせたい病院のチャットは、2 つの手段が両方要ります。用語と文体はファインチューニングで整えます。最新の指針は、ファインチューニングだけでは補えません。RAG で取得します。

手法の選び分けも場面で決まります。入力と出力の正解のペアが手元にあるなら SFT で、多くの顧客はここから始めます。好ましい応答と好ましくない応答の比較なら DPO です。RFT は最初の選択ではありません。報酬の信号で、より複雑な目標を最適化したい時に使います。問い合わせの分類のように用途を絞るなら、小型の GPT-4.1-mini で始めると、速くて低コストです。

モデルの種類と選び方

Foundry のモデルカタログには 1 万を超えるモデルがあり、毎月およそ 50 の新モデルが公開されます。種類は、基盤モデル・推論モデル・小規模言語モデル・マルチモーダル・ドメイン特化・業界モデルなどです。同じ系列の新モデルが出ると、旧モデルをずっと使えるとは限りません。

どのモデルでも、用途に合うかの判断を提供元に任せることはできません。モデルの説明やモデルカードを読んで選ぶのは、利用者の責任です。Azure 販売モデルの責任ある AI の審査も、リスクの緩和を保証しません。[13]

SAMPLE

トークンと費用と ROI [1.1.4]

トークンと課金の基本

トークンは、LLM がテキストを分けて処理する単位で、単語・文字の集まり・記号などです。英語ではおよそ 1 トークンが単語の 4 分の 3 です。[14]

価格は通常、入力トークンと出力トークンの合計で決まります。長いプロンプトは、トークンが増えて費用がかかり、応答に使える余地が減ります。推論モデルは精度が高い代わりに、遅延とトークン使用量、つまり費用が増えやすくなります。RAG でも、取得した文章が入力トークンを増やします。[15]

読み方の例です。同じ質問でも、社内文書の抜粋を 5 つ付けると、入力トークンが増えます。さらに推論モデルで考えさせると、考える分だけ出力側も増えます。精度が上がる代わりに、1 回あたりの費用と遅延が上がると読みます。

課金の型

課金の型	払い方	選ぶ条件
従量課金（Pay-as-you-go）	使った分だけ。長期契約なし。モデルは通常トークンで計測	開発・テスト・少量、変動が大きいトラフィック
コミットメント階層	一定の利用量に固定料金。超過分は超過レートで課金	コストを予測しやすくしたい時
プロビジョンド スループット（PTU）	配備した PTU ごとに時間単位。リクエストの有無は関係なし	本番で専用容量と安定した遅延が要る時
バッチ	Global Standard より低コスト。目標処理時間は 24 時間	即時の応答が要らない大量処理

SAMPLE

課金の型	払い方	選ぶ条件
Flex processing（プレビュー）	割引のトークン料金。リクエストごとに選ぶ	遅い応答や一時的な資源不足を許せる、低優先度の処理（モデルの評価など）
Priority processing	長期契約なしの従量課金で、priority の料金	時間に敏感で、一貫した低遅延が要る処理

プロビジョンド スループットは一時停止できません。課金が止まるのは、デプロイを削除した時だけです。使っていなくても課金が続くものは、プロビジョンドの配備と、時間課金のあるファインチューニング済みモデルで、存在する限り続きます。本番規模の大量利用でトークン課金が高くなる場合に、プロビジョンドが選択肢になります。[16]

場面で考えます。月末の締めに、数十万件の請求書の要約を一晩で流します。結果は翌朝に見ればよく、即時の応答は要りません。バッチが最も低コストです。逆に、顧客向けの窓口で、いつでも遅延を一定にしたいなら、プロビジョンドを選びます。

残りの 2 つの型も場面で見ます。毎月の利用量が読めて予算を確定したいなら、従量課金のままにする必要はありません。コミットメント階層です。固定料金に利用量が含まれ、超えた分だけ超過レートです。まだ利用量が読めない試作の段階や、日によって利用が大きく変わるアプリは、従量課金から始めます。長期の契約は要りません。使っていないのに課金が続く配備が残っていないかも、あわせて確かめます。

1 リクエストあたりの費用を下げる手段

次の 6 つが挙がっています。Model router、プロンプトキャッシュ、バッチ、プロビジョンド スループット、flex processing、Agent Optimizer です。このうち flex processing と Agent Optimizer はプレビューで、ほかは GA です。[17]

SAMPLE

ROI の考え方と FinOps

ROI（投資収益率）は、エージェントの価値が総コストを正当化するかを示します。拡大すべきものと、止めるべきものを分けるために使います。投入前に、事業の責任者が現行のプロセス、たとえば案件あたりの処理時間を測って、基準値（ベースライン）にします。基準値がなければ、改善の主張は信頼されません。

生産性の効果と、財務の効果は別です。時間の節約は、自動的には金銭の節約になりません。余った時間をどう使うかで、財務の価値が決まります。総コストは、導入と継続運用の費用で、ライセンスと消費課金、統合の作業、ガバナンスと監視の労力から成ります。[18]

場面で考えます。議事録の要約エージェントで、1 人あたり週に 2 時間の節約が見込めたとします。これは生産性の効果です。この時間で追加の案件をこなして売上が増えるか、人員の計画を変えて費用が減るかを決めるまでは、財務の効果は確定しません。ROI の計算でライセンス料だけを総コストに入れるのも誤りです。統合の作業と、ガバナンスと監視の労力も入れます。

FinOps は、クラウド費用の管理を、エンジニアリング・財務・事業部門が共同で行う実践です。ライフサイクルは、計画（Plan）、追跡（Track）、制御（Control）、最適化（Optimize）の順です。[19]

生成 AI の課題と緩和策 [1.1.5]

課題

課題	中身
ハルシネーション（捏造）	自信ありげに事実と異なる回答を出す。検証済みの事実を引くのではなく、もっともらしい文を作る

SAMPLE

課題	中身
学習データの期限	学習後の出来事を知らない（カットオフ）
非決定性	同じプロンプトでも呼び出しごとに応答が変わる。温度を 0 にしても完全には決定的にならない
記憶がない	各呼び出しは独立で、会話をコンテキストに含めない限り前の会話を覚えていない
計算と論理の誤り	精密な計算や形式論理で誤ることがある
バイアス	データの偏りがモデルの偏りになる
有害なコンテンツ	慎重な設計と緩和策がなければ、誤った内容や有害な内容を作りうる

根拠なし出力（Ungroundedness）は、提供された元の資料に無い、または元の資料と異なる、事実でない情報を出すことです。同一の入力で同一の出力に頼る設計は、してはいけません。[20]

信頼性と安全性は、責任ある AI の原則の 1 つで、AI システムが信頼性と一貫性をもって動くことを求めます。[21]

緩和策

課題	緩和策
捏造・根拠のない出力	RAG とグラウンディング。引用を有効にし、取得内容に沿うよう明確なシステムメッセージで指示する。検索などの外部機能で最新情報を得る
出力の誤り	結果を既知の事実やデータと突き合わせる自動検証。人による確認

SAMPLE

課題	緩和策
バイアス	多様で人口統計を代表する学習データ。人のレビューとフィードバックのループ
有害な内容	コンテンツの安全の仕組み（1.2.6 で扱う）

グラウンディングしても誤りは残ります。取得した内容があっても、モデルが不正確な回答を作ることがあるからです。検索が無関係または不完全な文章を返すと、グラウンディングしていても、回答は不完全または不正確になりえます。プロンプトを工夫しても、出力の検証は要ります。[22]

不均衡なデータはバイアスのあるモデルを作るので、代表的なサンプルを十分に集める設計が要ります。[23]

場面で考えます。社内規程の質問に答えるチャットが、存在しない条文を引用しました。原因はモデルの捏造です。再学習しても、規程に無い条文は消えません。規程を取得して根拠にし、引用を出させ、回答を元の資料と突き合わせる検証を加えます。それでも誤りは残りうるので、重要な回答は人が確認します。

間違えやすい点は 3 つです。温度を 0 にしても、完全に決定的にはなりません。RAG を入れても、捏造はゼロになりません。Azure 販売モデルの審査があっても、自社の文脈での評価は要ります。

社員の昇格の候補を提案する機能が、特定の部署の社員ばかりを挙げた場合も、手順は変わりません。学習データの偏りを疑い、代表的なデータを補い、人のレビューとフィードバックのループで偏りを見つけて直します。

SAMPLE

生成 AI が価値を出す場面 [1.1.6]

価値が出る場面

生成 AI が価値を出す典型は 3 つです。規模の拡大、自動化、個人の生産性です。

- コンテンツの起草・質問応答・要約・コード生成・従業員の業務支援
- エージェントによる自動化。依頼を解釈し、何をすべきか判断し、権限と統制の範囲内で連携システムを使って行動する
- スケーラビリティが要る処理。大規模・大量で、即時の応答が要らないジョブには、バッチ課金が向く

難しいのは、価値がどこで出て投資を正当化するのかの見極めです。[24]

価値を先に決める進め方

構築の前に価値を定義し、結果は決まった責任者（スポンサー）と定期的に見直します。初日からテレメトリを取る姿勢が、エージェントの価値を信頼できる形で説明するカギです。拡大の判断では、次の 3 つを見ます。使われているか、うまく機能しているか、拡大を正当化する価値を返しているかです。[25]

投資の判断

明確な戦略がないと、試行は価値を生みません。AI 投資は分断されます。リーダーが最初にやることは、AI が支えるべき事業の成果を定義し、優先事項へのスポンサーシップを示すことです。AI の戦略または憲章には、組織の優先事項と意思決定の権限を書き、AI 投資の価値をどう測るかも決めておきます。

組織全体の AI の方針と基準は中央が持ちます。AI Center of Excellence のような中央のガバナンス機能が責任ある AI の要件と共通の境界を定め、各部門が AI の機会を追う場は、その境界の外ではありません。承認された AI に

SAMPLE

従業員が広くアクセスでき、教育と組み合わさると、価値のある用途の発見と経験の蓄積が進みます。従業員は日々の業務を理解しているので、AI で改善できる機会を見つけられます。

データの基盤は、AI の導入前に全く新しいデータ戦略を作る必要はありません。必要なデータの能力は、採用する AI とユースケースで決まります。AI が必要とする組織のデータを特定し、安全にアクセスでき、適切にガバナンスできるようにします。[26]

場面で考えます。役員が「他社が入れたから」と全社に Copilot を配ると言い出しました。先にやるのは、導入が支えるべき事業の成果の定義と、現行の処理時間などの基準値の測定です。ここを飛ばすと、拡大するか止めるかを数字で決められません。

生成 AI が向く場面と、向かない場面を分けて覚えます。

場面	向き不向き
文書の起草・要約・質問応答・従業員の業務支援	向く
大量で即時の応答が要らない処理	向く（バッチ課金でスケールさせる）
依頼を解釈して連携システムを動かす業務	エージェントで自動化できる（権限と統制の範囲内）
数値の予測・分類・異常検知	従来型 AI（機械学習）が向く
精密な計算や形式論理が結果を決める処理	生成 AI 単独では誤ることがある。検証と組み合わせる

試行の段階から拡大の段階へ進む前の問いは、3 つあります。使われているか、うまく機能しているか、拡大を正当化する価値を返しているかです。3 つ目は、導入前の基準値と初日からのテレメトリが無いと、答えられません。

SAMPLE

プロンプト エンジニアリングが出力に与える影響 [1.2.1]

影響の中身

プロンプトは、モデルがタスクを終える助けになります。ただし科学というより技芸で、よいプロンプトを作るには経験と直感が要ります。プロンプトが何であれ、モデルは最もありそうだと判断した続きを返しているだけです。[27]

出力を動かす要素は次の 5 つです。

要素	出力への影響
指示の順序	情報を置く順序で出力が変わる
システム メッセージ	チャットでは指示をシステム メッセージで渡す。フューショットの例は、ユーザーとアシスタントのやり取りの形で示す
温度（Temperature）	出力のランダム性を制御する。創作は高め、法的文書などは低めが推奨される
グラウンディング データ	最新で信頼できる情報が要る用途で、回答の元になるデータを与えると信頼性が上がる
モデルの種類	モデルごとに挙動が違う。ある手法の学びが、ほかのモデルに当てはまらないことがある

RAG では、システム メッセージとプロンプトが、取得した内容をモデルがどう使うかを導く指示になります。[28]

限界とリスク

プロンプトの工夫には限りがあります。次の 4 つを押さえます。

SAMPLE

- フューショットの例は、モデルを恒久的に学習させません。その推論の間だけ、望む応答に寄せます。モデルそのものを変えたいなら、ファインチューニングです。
- 入力サイズには上限があります。世代ごとに増えても上限はあり、限られた空間を効率よく使う必要があります。
- プロンプトの手法は、gpt-5 や o シリーズのような推論モデルには推奨されません。
- プロンプトインジェクションでは、ユーザーがシステムレベルの制限を書き換え、いわゆる「脱獄」を起こすことがあります。

守りにもプロンプトを使えます。メタプロンプトやシステム メッセージで、モデルの役割・能力・限界を限定し、プロンプトを書き換える試みを無視させる設計が推奨されます。ただしこれだけでは守りきれないので、1.2.6 の仕組みと組み合わせます。[29]

場面で考えます。社内向けのチャットが、長い規程を丸ごと入力に詰めて、後半の条文を無視した回答を返しました。入力サイズの上限と、情報の順序が原因です。必要な部分だけを取り出して与え、指示を先頭に置き、末尾でもう一度繰り返します。モデルを学習し直しても解決しません。

温度の使い分けも場面で決まります。製品のキャッチコピーを何案も出させたいなら、温度は高めで、案がばらけるほうが役に立ちます。契約書の条文の要約のように、正確さが要る用途は、温度を低めにします。ただし温度を下げても、同じ入力で同じ出力になるとは限りません。

もう 1 つ、推論モデルを使う場合の注意があります。手法の学びがモデルごとに違うため、非推論モデルで効いた書き方を、gpt-5 や o シリーズへそのまま持ち込むのは推奨されません。

SAMPLE

プロンプト エンジニアリングの手法 [1.2.2]

プロンプトの 4 つの構成要素

構成要素	役割
指示（Instructions）	最もよく使う要素。モデルにやらせることを伝える
主要な内容（Primary content）	モデルが処理・変換する対象のテキスト。翻訳元の文章など
手がかり（Cue）	出力の書き出しの起点となる接頭辞。望む出力へ導く
補足の内容（Supporting content）	現在の日付・ユーザー名・好みなど、出力に影響を与える背景情報

翻訳の依頼なら、「次の文を英語にしてください」が指示、翻訳元の文章が主要な内容です。[30]

技法

技法	中身
フューショット（Few-shot learning）	プロンプトに学習例を含めて、新しいタスクに適応させる
指示を先頭に置く	やらせたいタスクを最初に示し、その後に文脈や例を置く
指示を末尾で繰り返す	末尾の情報の影響が大きい（recency bias）ため、有効な場合がある
出力を誘導する（Prime the output）	末尾に数語を入れて、箇条書きなど望む形式に導く

SAMPLE

技法	中身
タスクの分解	小さな手順に分けたほうが、性能が上がることが多い。事実の主張が多い大きな文章では、分けると大きな差が出る
思考の連鎖（Chain of thought）	一歩ずつ進め、全ての手順を示させる。誤りが減り、評価しやすくなる
グラウンディングの文脈を与える	最新で信頼できる情報が要る用途で強く推奨される
具体的に書く	解釈の余地を最小にし、動作の範囲を絞る
逃げ道を与える	答えが無い時の代替を指示する。たとえば「見つからない」と答えさせる

思考の連鎖は、推論モデルでないモデルにだけ使えます。推論モデルはもともと段階的に考えるためです。逃げ道の指示は、答えが無い時に誤った回答を作るのを避ける効果があります。表形式のデータは GPT が理解しやすく、トークンも節約できます。[31]

間違えやすいのは、手法の目的です。フューショットは出力の形と調子を見せる手法で、知識を足す手法ではありません。思考の連鎖は理由を見せる手法で、最新情報を足す手法ではありません。

場面で考えます。顧客の問い合わせメールを「要望・不具合・その他」に分類させたところ、ラベルの表記がばらつきました。分類の例を 3 件、入力と出力のペアで見せるフューショットが効きます。さらに、出力の書き出しに「分類:」と添えると、形式が揃います。

場面をもう 1 つ。複雑な依頼を 1 つのプロンプトにまとめたら、回答が雑になりました。この場合は、タスクを小さな手順に分けて、1 つずつ依頼します。「資

SAMPLE

料を読む、論点を挙げる、優先順位を付ける、報告文に整える」の順です。手順を小さく分けたほうが LLM の性能が上がる例は、少なくありません。

回答が見つからない時に、もっともらしい作り話を返す場合は、逃げ道を与えます。「資料に書かれていなければ『資料に記載がありません』と答える」と指示すれば、誤った回答を作るのを避けやすくなります。ただし、指示だけで誤りがゼロになるわけではありません。

グラウンディングの要件 [1.2.3]

グラウンディングが要る業務

グラウンディングデータは、モデルに推測を減らさせるために与える、取得した内容です。LLM は学習時点で公開されていたデータで学習されているため、社内の資料や最新の情報には足りません。社内の非公開データや頻繁に変わる情報に基づく回答が要る時に、RAG で根拠を与えます。[32]

要件は 3 つです。

- 最新の社内情報を答えに使う
- 回答の根拠を示す（引用）
- アクセス制御を守る。エージェントは、現在のユーザーのアクセス権の範囲のデータを受け取る[33]

品質と安全

RAG の品質は、コンテンツの準備・検索の設定・プロンプトの設計で決まります。

安全の面では、検索の時点でアクセス制御を効かせる、権限の効いた検索にします。Foundry IQ では、ドキュメント単位のセキュリティ フィルターを使いま

SAMPLE

す。元のコンテンツへのアクセスを管理しないまま、グラウンディングされた回答を出してはいけません。索引内の機密情報を漏らすからです。取得した内容は、信頼できない入力として扱います。文書や取得した文章に仕込まれたプロンプトインジェクションの危険を下げるためです。

Microsoft Copilot の検索の範囲は、Restricted SharePoint Search で、指定したサイトに一時的に限定できます。データのガバナンスが整うまでの手当てで、恒久の対策ではありません。[34][35]

根拠性（Groundedness）のある回答は、与えられた情報に忠実で、推測や捏造を避けます。Groundedness detection は、LLM の応答が提供した元の資料に基づいているかを検出して、捏造のリスクを下げます。モードは 2 つあります。

モード	特徴	向く場面
Non-Reasoning	高速で、結果は grounded か ungrounded の 2 値	遅延が重要なリアルタイム用途
Reasoning	根拠のない箇所の説明が付く	開発やデバッグで原因を知りたい時

根拠性の検出は、文書の埋め込みと整形なしには使えません。

groundedness detection（Azure AI Content Safety と Azure OpenAI の根拠性の検出の機能）は、プレビューの機能で、2027-04-30 に廃止が予定されています。代わりに、Foundry の評価ツールか、LLM やシステム プロンプトを使った自前の根拠の検証を検討します。[36]

データの基盤の要件は、AI が必要とする組織のデータを特定し、安全にアクセスでき、適切にガバナンスできるようにすることです。[37]

SAMPLE

場面で考えます。人事部が、評価面談の記録を検索して回答するチャットを作ります。一般社員が、他人の評価を引き出せてはいけません。検索の時点で、質問した人のアクセス権でフィルターします。回答の後に隠しても、索引の中の情報は漏れます。

根拠性のモードの選び分けも場面で決まります。顧客向けの窓口は、応答の遅延を抑える必要があるので、Reasoning は選びません。高速な Non-Reasoning です。開発の段階で、なぜ根拠がないと判定されたのかを調べたいなら、説明が付く Reasoning です。

全社の展開の前に、検索の範囲を絞りたい場合は、Restricted SharePoint Search で、Microsoft Copilot の検索を、指定したサイトに一時的に限定します。限定は一時的な手当てにすぎないので、これだけでは共有の問題は解決しません。並行して、共有の見直しと秘密度ラベルを進めます。

RAG の使い方 [1.2.4]

RAG の流れ

RAG（検索拡張生成）は、検索と LLM を組み合わせて、応答を自社データに根拠づけるパターンです。3 つの段階を順に踏みます。[38]

順	段階	中身
1	取得（Retrieve）	質問に関連する内容を、インデックスやデータストアから検索する
2	拡張（Augment）	質問と取得した内容（グラウンディングデータ）を合わせて、プロンプトにする

SAMPLE

順	段階	中身
3	生成（Generate）	拡張されたプロンプトを受けたモデルが、取得内容に根拠を置いた回答を生成する。不正確さが減り、引用もできる

用語	中身
インデックス	検索に最適化されたデータ構造。検索の型は、キーワード、セマンティック、ベクトル、ハイブリッド
エンベディング（埋め込み）	ベクトル類似度検索に使う、コンテンツの数値表現
Foundry IQ（Azure AI Search）	RAG で推奨されるインデックスのストア

Foundry での導入の順

1. データの準備: 非公開の文書や知識ベースを整理し、検索できる単位に分割（チャンク化）する
2. インデックスの設定: Foundry IQ のインデックスなどを作る
3. Foundry への接続: プロジェクトからインデックスへ接続を作る
4. RAG アプリの構築: 検索をモデルの呼び出しに組み込む
5. テストと評価: 検索の品質がよく、回答が正確で、適切に引用されているかを確かめる

判断の勘所

道具は 1 つに決め打ちしません。選び方は 3 つです。非公開や頻繁に変わるデータに根拠づけた回答には RAG、ふるまいや文体の変更にはファインチュ

SAMPLE

ーニング、エージェントが検索を必要とするならエージェントのツールとして検索を使い、検索の仕組みまで完全に制御して RAG アプリを組むなら Foundry SDK を使います。

Agentic RAG（agentic retrieval）は、従来の RAG を進めた形です（文書の言葉は evolution）。モデルが複雑な入力を複数のサブクエリに分け、並列に実行して、構造化されたグラウンディングデータを返します。

トラブルの切り分けは次のとおりです。

- 関連する文章が返らない: チャンクの分け方、埋め込みモデルの質、検索の設定（キーワード・セマンティック・ハイブリッド）を見直す
- 取得した文章が多すぎてトークンの上限をすぐ使い切る: フィルタ・ランキング・要約で絞る
- 情報漏えい: アクセスとプロンプトを慎重に設計しないと、RAG のシステムは機密コンテンツを露出させる

場面で考えます。社内のナレッジの検索チャットで、質問によって無関係な文章ばかり返ります。モデルを替える前に、チャンクの分け方と検索の型を見直します。それでも足りない時に、評価の結果を見て、埋め込みモデルを替えます。[39]

Agentic RAG が向くのは、1 つの質問に複数の観点が含まれる場合です。「A 製品の保証期間と、B 製品との違いを教えて」のような入力を、モデルがサブクエリに分けて並列に検索し、構造化されたグラウンディングデータとして返します。単純な事実の検索に、Agentic RAG は要りません。

RAG の評価では、検索の品質と回答の品質を分けて見ます。関連する文章が取れているのに回答が悪いなら、プロンプトとシステム メッセージを見直しま

SAMPLE

す。そもそも文章が取れていないなら、プロンプトを直しても効きません。チャンクと埋め込み、検索の設定を見ます。

データが AI に与える影響 [1.2.5]

データの種類と役割

呼び名	役割
ソースデータ	特定の時点の観測データ。パイプラインへの入力の候補で、ラベルを付けて使える
学習データ	ソースデータの一部。モデルにサンプルを与える。無いとモデルは関連する出力を作れない
評価データ	ソースデータの一部。学習中の性能を監視・検証し、ハイパーパラメータの調整の指針にする
テストデータ	学習に使っていないデータ。学習済みモデルの予測力を検証する

データの型は 3 つです。

型	例
構造化	データベース、スプレッドシート
半構造化	JSON、XML
非構造化	文書、画像

マルチモーダルデータは、テキスト・画像・音声・動画などの複数の種類を、同じ学習データセットで組み合わせたものです。

SAMPLE

教師あり学習では、ラベル付きのサンプルをモデルに与えます。望む結果が明確でない場合は向きません。2 値分類の学習データには、該当する例と該当しない例の両方が要ります。[40]

品質と代表性

- データの偏り: 不均衡なデータはバイアスのあるモデルを作る。少数のデータのオーバーサンプリングなどを検討する。特徴の分布が偏ると、過小に代表されるグループのデータ点を誤って予測しうる
- 見える化: Responsible AI ダッシュボードのデータ分析で、過大・過小の代表の問題とデータの分布を可視化できる[41]
- ノイズ: 取り除き、各行が用途に関連する明確な観測になるようにする
- 重複: 冗長なデータを除くと、学習データの正確さと代表性が保たれる
- 個人データ: 匿名化では得られない予測力に不可欠でない限り、学習データから除く
- 古いデータ: データドリフトとコンセプトドリフトを監視し、新しい観測で学習データを更新する

本番のデータを学習データに使う場合は、元のデータ分類の制約が全部適用されます。データ設計は、統計的な実験に基づく反復のプロセスです。[42]

公平性の原則は、AI システムが全員を公平に扱い、似た条件の集団に異なる影響を与えないことを求めます。[43]

場面で考えます。採用の選考を支援するモデルが、過去 10 年の採用データで学習し、特定の属性の応募者を低く評価しました。原因は、過去の採用が偏っていたことです。モデルの構造を疑う前に、学習データの代表性を確かめ、不足するグループを補い、公平性を評価します。

SAMPLE

症状から原因を引く練習です。

症状	疑う原因	対処
特定のグループへの予測だけ外れる	データの分布の偏り・過小な代表	少数のデータを補う（オーバーサンプリング）・データ分析で可視化
学習に使ったデータでの成績だけで、モデルの良し悪しを決めている	検証の仕方（学習に使っていないデータで確かめていない）	学習に使っていないテストデータで、学習済みモデルの予測力を検証する
運用して時間が経つと精度が落ちる	データドリフト・コンセプトドリフト・古いデータ	監視し、新しい観測で学習データを更新する
同じ記録が多数ある	重複	重複排除
予測に不要な個人データが入っている	個人データ	匿名化できない予測力に不可欠でない限り除く

2 値分類のモデルで「不正な取引」だけを集めて学習させるのは誤りで、それでは分類を学べません。「不正でない取引」の例も要ります。

安全な AI の重要性 [1.2.6]

責任ある AI の 6 原則

Microsoft の Responsible AI Standard は、6 つの原則でできています。

原則	求めること
公平性（Fairness）	全員を公平に扱い、似た条件の集団に異なる影響を与えない（包括性とあわせて扱われる）
信頼性と安全性	設計どおりに動き、想定外の状況に安全に対応し、悪意ある操作に耐える

SAMPLE

原則	求めること
プライバシーとセキュリティ	正確な予測にデータが要るため、プライバシーとデータ保護に特に注意する
包括性（Inclusiveness）	AI システムが世界の人々に力を与え、関わりを促す（Azure Machine Learning の文書では公平性とあわせて扱われる）[44]
透明性	AI の挙動に有用な説明を与える解釈可能性が、重要な要素
説明責任	AI を設計・展開する人が、システムの動作に責任を負う

安全な AI が要る理由は、AI が正確な予測のためにデータを必要とし、AI に固有の攻撃を受け、設計が甘いと有害な内容を出すからです。6 原則のうち「信頼性と安全性」と「プライバシーとセキュリティ」に直結します。[45]

AI への攻撃

AI は、AI に固有の攻撃を受けます。

- プロンプトインジェクション: ユーザーの入力で、システムの指示を書き換える
- 脱獄攻撃: モデルの制限を外させる
- データポイズニング: 事前学習・ファインチューニング・埋め込みの各段階でデータが操作され、脆弱性・バックドア・バイアスが入り込む

間接プロンプトインジェクションは、LLM が Web サイトやファイルなど外部のソースから入力を受ける時に、攻撃者が外部コンテンツに指示を埋め込むものです。[46]

SAMPLE

守りの道具

Azure AI Content Safety は、モデル・エージェント・サービスの中の、ユーザー生成と AI 生成の有害なコンテンツを検出する AI サービスです。[47]

機能	中身
有害コンテンツの検出	テキストと画像で、性的な内容、暴力、憎悪、自傷を、多段階の重大度で検出する
Prompt Shields	LLM へのユーザー入力による攻撃のリスクを、テキストでスキャンする
ガードレール	リスクの種類、スキャンする箇所、検出した時の応答を指定する、制御の集まり
Foundry の中のガードレールの無料枠	最初の 1000 文字までのテキストのモデレーションが無料

攻撃・リスク	守り
ユーザー入力による攻撃（プロンプトインジェクション・脱獄）	Prompt Shields
有害な出力	Content Safety のカテゴリ検出とガードレール（コンテンツフィルター）
根拠のない出力	Groundedness detection

安全の進め方

AI のセキュリティは、3 段階で進めます。AI のアプリとデータを発見（Discover）し、保護（Protect）し、コンプライアンスのために統制（Govern）します。[48] Foundry の責任ある AI も、同じ 3 段階です。[49]

出発点は、AI 専用でない基盤の保護です。ID とデバイスのアクセス、データ保護、脅威対策を抜きにして、AI 固有の機能だけで守ることはできません。最初

SAMPLE

にやるのは、Microsoft Entra ID による多要素認証と条件付きアクセスで、環境を守る最も強力な手段の 1 つです。Microsoft Defender の脅威対策は、クラウド環境全体の AI アプリとモデルの発見と統制を含みます。[50]

場面で考えます。顧客向けのチャットが、入力に「これまでの指示を無視して」と書かれて、社内向けの手順を話し始めました。入力への攻撃なので、Prompt Shields で入力をスキャンし、ガードレールで検出時の応答を決めます。システム メッセージの強化だけでは足りません。基盤に MFA と条件付きアクセスが無いなら、そちらが先です。

3 段階の使い方の例です。

段階	例
発見（Discover）	社内でどの AI アプリが使われ、どのデータに触れているかを把握する。Foundry ではリスクの発見で、たとえば敵対的なプロンプトでエージェントを試し、脱獄の弱点を洗い出す
保護（Protect）	出力と実行時の保護。コンテンツフィルターとガードレールで、有害な出力がユーザーに届く前にブロックする。データを守る
統制（Govern）	トレースと監視のツールで、エージェントの挙動を追う。コンプライアンスの連携で要件に合わせる

外部から取り込んだデータで学習やファインチューニングをする場合は、データポイズニングを疑います。混ぜられたデータで、モデルに脆弱性やバックドア、偏りが入り込むからです。学習の段階に限りません。埋め込みを作る段階でも起こります。

SAMPLE

機械学習が価値を出す場面 [1.2.7]

機械学習の用途

機械学習（従来型 AI）は、予測・分類・異常検知のように、定められた目的でデータを分析する場面で価値が出ます。モデルを学習させる目的は、新しい未知の状況を、分類・予測できるようにすることです。[51]

問題のタイプ	中身	例
分類	新しいデータがどのカテゴリに属するかを予測する	不正検知、手書き文字の認識、物体検出
回帰	独立変数から数値の出力を予測する	価格
時系列予測（Forecasting）	時間とともに変わる値を予測する	収益・在庫・売上・需要
コンピュータビジョン	画像データで学習する	画像分類、物体検出
自然言語処理（NLP）	言語のデータを扱う	テキストの分類

AutoML で最初にやるのは、問題を分類・予測・回帰・コンピュータビジョン・NLP のどれかとして特定することです。[52]MLOps v2 の文書には、アーキテクチャが 3 系統（古典的な機械学習、コンピュータビジョン、自然言語処理）示されています。[53]

生成 AI でなく機械学習を選ぶ判断

「来月の店舗別の売上を予測したい」「取引の不正を検知したい」は、数値の予測と分類です。生成 AI の出番ではなく、機械学習です。文章を作る、要約する、質問に答えるが目的なら、生成 AI です。

SAMPLE

AutoML は、目標の指標を指定して、Azure Machine Learning にモデルの学習とチューニングを任せたい時に使います。データサイエンスの専門性は問いません。

推論の形は 2 つです。

推論の形	中身	場面の例
バッチ推論	非同期で並列に処理し、結果を後で分析する	夜間にまとめて全顧客の離反を予測する
リアルタイム推論	HTTPS でほぼ即時に応答を得る	申込みの画面でその場で審査する

場面を 3 つ見ます。

場面	問題のタイプ
倉庫の画像から傷のある部品を見分ける	コンピュータビジョンの分類
過去の売上から来四半期の需要を出す	時系列予測
物件の特徴から家賃を出す	回帰

いずれの場面にも、生成 AI を選ぶ理由はありません。

製品のレビューの感想を分類したい場合は、言語のデータなので NLP の問題ですが、新しい文章を作るわけではありません。「言語を扱う」ことだけで生成 AI と決めないでください。

Azure Machine Learning は、機械学習プロジェクトのライフサイクルを加速し、管理するクラウド サービスです。MLOps を実践し、モデルを安全で監査できる本番環境に出したい個人とチーム向けです。[54]

SAMPLE

機械学習のライフサイクル [1.2.8]

一生の順序

機械学習は、まず目的とゴールを持つプロジェクトとして始まり、データ・アルゴリズム・モデルの実験を繰り返します。MLOps v2 のライフサイクルは、1 本の流れではありません。内側ループと外側ループに分かれます。

ループ	順序
内側ループ（モデル開発）	データの取り込み → 探索的なデータ分析 → 実験 → モデルの開発と評価 → 本番用のモデルの登録
外側ループ（本番化）	本番前のステージングとテスト → 本番デプロイ → モデル・データ・インフラの監視

最初のステップは、管理と設定で、リソース、ロール、リポジトリ、CI/CD パイプライン、監視を作ります。本番にデプロイできるモデルは、機械学習ワークスペースのレジストリに登録します。ステージングとテストでは、本番データでの再学習とテスト、エンドポイントの性能の試験、データ品質のチェック、単体テスト、データとモデルのバイアスなど責任ある AI のチェックをします。これに合格しないモデルは、本番へ昇格できません。合格したモデルは、人の承認（human-in-the-loop）の関門を経て昇格させることができます。[53]

MLOps

MLOps は、DevOps の原則（継続的インテグレーション・デプロイ・デリバリー）を機械学習のライフサイクルに適用したものです。効果は、実験とモデル開発の高速化、本番デプロイの高速化、品質保証とエンドツーエンドの系譜の追跡の向上です。ツールは、モデルの監視・再学習・再デプロイを助けます。

SAMPLE

- バージョン管理: 登録したモデルは名前とバージョンで識別され、同じ名前で登録するとバージョン番号が上がる
- 段階的な展開（ロールアウト）: 同一のエンドポイントの複数のデプロイにトラフィックを振り分け、A/B テストができる
- 監査: 誰がモデルを公開し、なぜ変更し、いつ本番に展開したかの系譜（リネージ）のデータを記録する。モデルのライフサイクルは、再現できないとしても、監査できなければならない
- データドリフトの監視: 入力データの分布が、学習データや直近の本番データからどう変わったかを見る。再学習を起こす条件と更新頻度を、先に決めておく[55]

ドリフトには 2 種類あります。データドリフトは、入力データが変わって、モデルが古くなることです。コンセプトドリフトは、外部の条件が変わって、モデルの予測が現実を反映しなくなることです。[56]どちらも、新しい観測で学習データを更新せずに放置すると、精度は戻りません。

主な役割は次の 4 つです（文書は、データ アナリスト、プロジェクト リード、プロジェクト／プロダクト オーナー、プラットフォームの技術サポートなども挙げます）。

役割	担当
データエンジニア	データ資産（データ基盤）の主な所有者
データサイエンティストと機械学習エンジニア	内側ループ（モデル開発）
事業のステークホルダー（例: マーケティング担当）	プロジェクトに関わり、データの所有に応じてワークスペースの責任を負う
モデルテスター	テスト・ステージング環境で試験し、CI/CD から機能を分離する

SAMPLE

場面で考えます。本番に出した需要予測が、半年後から外れ始めました。商品の構成が変わり、入力の分布が学習時から動いたことが疑われます。データドリフトの監視で確かめ、事前に決めた条件で再学習し、ステージングで試験してから、新しいバージョンを登録して段階的に展開します。

順序の問題は、前後の関係で解きます。

手	位置
登録	内側ループの最後。モデルの開発と評価が済んだ後
ステージングとテスト	本番デプロイの前
人の承認	ステージングとテストに合格した後で、本番への昇格の前
監視	本番デプロイの後。その結果が再学習の引き金になる

この前後の関係は、並べ替えの形でも問われ、順序を入れ替えると成り立ちません。

新しいバージョンの安全な展開では、同じエンドポイントの中に旧バージョンと新バージョンの 2 つのデプロイを置いて、トラフィックの一部だけを新しいほうへ振り分けます。結果を比べて、問題がなければ割合を増やします。

SAMPLE

AI システムのセキュリティ [1.2.9]

認証とアクセス

方式	特徴
Microsoft Entra ID	RBAC のロールが適用される。安全性が高く、きめ細かなアクセス制御ができる。推奨される
キー認証	ロールの制限なしに全アクセスが得られる。開発には便利だが、本番では推奨されない

Foundry Tools の認証は、リソースキー、トークン、Microsoft Entra ID の 3 通りです。[57]本番で優先するのは Entra ID で、キーは推奨されません。Entra ID で RBAC を実装し、必要最小限のアクセスだけを与えるのが最小権限の原則です。[58]

エージェントを呼び出すだけで、作成も変更もしない主体には、最小権限の Foundry Agent Consumer を割り当てます。Foundry User・Owner・Account Owner・Project Manager のロールは、旧名（Azure AI User など）から改名されたもので、ロール ID と権限は変わっていません。[59]

LLM がバックエンドのシステムにアクセスする権限も、最小限にします。[60]

アプリとデータの守り

守り	中身
暗号化	組織のコンテンツは保存時と転送時に暗号化される。HTTP ベースの通信は TLS
監査ログ	アクセスと利用を定期的に監視し、データアクセスと変更を追跡する詳細な監査ログを保持する

SAMPLE

守り	中身
外部コンテンツの分離	外部コンテンツをユーザーのプロンプトから分離し、信頼できないコンテンツの影響を限定する
人の承認	高リスクな操作には人の承認（human in the loop）を使う
秘密の保管	Azure Machine Learning の連携には VNet・ネットワーク セキュリティ グループ・Key Vault（アクセス情報などの秘密の保管）がある
DSPM for AI（Microsoft Purview）	過剰共有の場所と、秘密度ラベル・DLP ポリシーの適用漏れを把握する
Defender for Cloud Apps	新しい AI アプリの利用を通知し、リスクスコアを計算し、許可やブロックができる
Defender for Foundry Tools	AI ワークロードで脅威が見つかると、セキュリティ アラートを生成する

暗号化と監査ログの守りは、Copilot Studio の責任ある AI の文書が挙げるものです。[61] Defender for Foundry Tools のアラートは Foundry の文書が、DSPM for AI と Defender for Cloud Apps は AI のセキュリティの文書が説明しています。[35]

AI システムは、データの収集・利用・保管について透明性を求める、プライバシー法に従う必要があります。[62]

場面で考えます。経理部が、請求書の処理を代行するエージェントを Foundry で作ります。開発の間は API キーで動かしています。本番では、キーを Entra ID の認証に替え、エージェントには呼び出しだけに必要な最小のロールを割り当てます。支払いの実行のような高リスクな操作を、エージェント

SAMPLE

だけで完結させてはいけません。人が承認する流れにします。受け取るメールや添付ファイルの指示を、そのまま実行してはいけません。信頼できない外部コンテンツとして分けて扱います。

ロールの選び分けの例です。エージェントの呼び出しだけをするアプリには Foundry Agent Consumer を割り当てます。作成や変更もする開発者には、開発者向けの最小権限ロールの Foundry User が要ります。管理まで担う人でなければ、Project Manager や Owner は割り当てません。

利用状況の把握では、まず DSPM for AI で過剰共有の場所と、秘密度ラベルや DLP ポリシーの適用漏れを見ます。次に、従業員が使い始めた新しい AI アプリは、Defender for Cloud Apps が検出し、リスクスコアを計算して、許可かブロックを決められます。Foundry の AI ワークロードで脅威が見つかれば、Defender for Foundry Tools がアラートを生成します。

取り違えやすい組み合わせ [1.1.3/1.1.4/1.2.1/1.2.3/1.2.6/1.2.8/1.2.9]

組み合わせ	違い
ファインチューニングと RAG	前者はふるまい・文体・タスク性能を変える。後者は新しい知識や変わる情報を、取得して根拠にする
フューショットとファインチューニング	前者はその推論の間だけ。後者はモデルを追加学習で変える
プロンプトインジェクションとデータポイズニング	前者は入力の時の攻撃。後者は事前学習・ファインチューニング・埋め込みの段階のデータへの攻撃
グラウンディングと根拠性の検出	前者は根拠となる内容を与えること。後者は応答が元の資料に基づくかを検出すること

SAMPLE

組み合わせ	違い
Prompt Shields と Groundedness detection	前者はユーザー入力による攻撃を見る。後者は応答が元の資料と合うかを見る
バッチ課金とバッチ推論	前者は即時の応答が要らない大量処理の課金（24 時間が目標）。後者は機械学習の非同期の推論の形
従量課金とプロビジョンド スループット	前者は使った分だけ。後者は配備した分を時間単位で払い、使わなくても課金される
生産性の効果と財務の効果	時間の節約は、その時間の使い方が決まるまで金銭の節約ではない
Microsoft Copilot と Copilot Chat	前者は仕事の文脈の AI でライセンスが費用の出どころ。後者は主に Web データのチャットで無料
Copilot Studio と Agent 365	前者はエージェントを作る。後者は複数のプラットフォームにまたがるエージェントを、中央で可視化し統制する
Foundry と Foundry Tools	前者はカスタムなエージェントやアプリを作るプラットフォーム。後者は既製のモデルやサービスで、Foundry から API で使う
Entra ID 認証とキー認証	前者は RBAC のロールが効く。後者はロールの制限なしに全アクセスが得られる
データドリフトとコンセプトドリフト	前者は入力データの変化。後者は外部の条件の変化で、予測が現実を反映しなくなること

次の 3 つは、数字と名前がそのまま問われます。

- ファインチューニングのデータ量: 初期テストが 50〜100 件、本番が 500 件以上

SAMPLE

- Groundedness detection の廃止予定日: 2027-04-30
- Foundry の中のガードレールの無料枠: 最初の 1000 文字までのテキスト

SAMPLE

サンプルはここまでです

続き(第2章〜巻末)は、AB-731 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。

Footnotes

1. https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai/strategy ↩
2. https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai/strategy ↩
3. https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/mlops-technical-paper ↩
4. https://learn.microsoft.com/en-us/azure/well-architected/ai/training-data-design ↩
5. https://learn.microsoft.com/en-us/dotnet/ai/conceptual/how-genai-and-llms-work ↩
6. https://learn.microsoft.com/en-us/agent-framework/journey/llm-fundamentals#what-is-an-llm ↩
7. https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai/strategy ↩
8. https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai/strategy ↩

SAMPLE

9. https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai/strategy ↩
10. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/foundry-models-overview#choosing-between-models-sold-by-azure-and-models-from-partners-and-community ↩
11. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/fine-tuning-overview#top-use-cases-for-fine-tuning ↩
12. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/retrieval-augmented-generation#choose-an-approach-in-foundry ↩
13. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/foundry-models-overview ↩
14. https://learn.microsoft.com/en-us/dotnet/ai/conceptual/understanding-tokens ↩
15. https://learn.microsoft.com/en-us/agent-framework/journey/llm-fundamentals#tokens-the-building-blocks ↩
16. https://learn.microsoft.com/en-us/azure/foundry/concepts/billing-concepts ↩
17. https://learn.microsoft.com/en-us/azure/ai-foundry/openai/how-to/manage-costs ↩
18. https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai-agents/business-strategy-plan#3-return-on-

SAMPLE

investment-roi ↩

19. https://learn.microsoft.com/en-us/azure/ai-foundry/openai/how-to/manage-costs ↩
20. https://learn.microsoft.com/en-us/agent-framework/journey/llm-fundamentals#what-llms-struggle-with ↩
21. https://learn.microsoft.com/en-us/azure/machine-learning/concept-responsible-ai#reliability-and-safety ↩
22. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/retrieval-augmented-generation#limitations-and-troubleshooting ↩
23. https://learn.microsoft.com/en-us/azure/well-architected/ai/training-data-design ↩
24. https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai/strategy ↩
25. https://learn.microsoft.com/en-us/microsoft-copilot-studio/guidance/agent-business-value-overview ↩
26. https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai/strategy ↩
27. https://learn.microsoft.com/en-us/azure/ai-foundry/openai/concepts/prompt-engineering#basics ↩
28. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/retrieval-augmented-generation ↩

SAMPLE

29. https://learn.microsoft.com/en-us/ai/playbook/technology-guidance/generative-ai/mlops-in-openai/security/security-plan-llm-application ↩

30. https://learn.microsoft.com/en-us/azure/ai-foundry/openai/concepts/prompt-engineering#prompt-components ↩

31. https://learn.microsoft.com/en-us/azure/ai-foundry/openai/concepts/prompt-engineering#scenario-specific-guidance ↩

32. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/retrieval-augmented-generation#what-is-rag ↩

33. https://learn.microsoft.com/en-us/microsoft-copilot-studio/guidance/responsible-ai#data-privacy-and-security ↩

34. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/retrieval-augmented-generation#security-and-privacy-considerations ↩

35. https://learn.microsoft.com/en-us/security/security-for-ai/protect ↩ ↩2

36. https://learn.microsoft.com/en-us/azure/foundry/openai/concepts/content-filter-groundedness ↩

37. https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai/strategy ↩

SAMPLE

38. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/retrieval-augmented-generation#how-does-rag-work ↩

39. https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/retrieval-augmented-generation#rag-workflow-in-foundry ↩

40. https://learn.microsoft.com/en-us/azure/well-architected/ai/training-data-design#types-of-data ↩

41. https://learn.microsoft.com/en-us/azure/machine-learning/concept-data-analysis ↩

42. https://learn.microsoft.com/en-us/azure/well-architected/ai/training-data-design#data-preprocessing ↩

43. https://learn.microsoft.com/en-us/azure/machine-learning/concept-responsible-ai ↩

44. https://support.microsoft.com/en-us/privacy/what-is-responsible-ai（6 原則と Responsible AI Standard） ↩

45. https://learn.microsoft.com/en-us/azure/machine-learning/concept-responsible-ai ↩

46. https://learn.microsoft.com/en-us/ai/playbook/technology-guidance/generative-ai/mlops-in-openai/security/security-plan-llm-application ↩

47. https://learn.microsoft.com/en-us/azure/ai-services/content-safety/overview ↩

SAMPLE

48. https://learn.microsoft.com/en-us/security/security-for-ai/ ↩

49. https://learn.microsoft.com/en-us/azure/ai-foundry/responsible-use-of-ai-overview ↩

50. https://learn.microsoft.com/en-us/security/security-for-ai/prepare ↩

51. https://learn.microsoft.com/en-us/azure/well-architected/ai/training-data-design ↩

52. https://learn.microsoft.com/en-us/azure/machine-learning/concept-automated-ml#when-to-use-automl-classification-regression-forecasting-computer-vision-and-nlp ↩

53. https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/mlops-technical-paper#architecture ↩ ↩2

54. https://learn.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-machine-learning ↩

55. https://learn.microsoft.com/en-us/azure/machine-learning/concept-model-management-and-deployment ↩

56. https://learn.microsoft.com/en-us/azure/well-architected/ai/test ↩

57. https://learn.microsoft.com/en-us/azure/ai-services/authentication ↩

58. https://learn.microsoft.com/en-us/azure/foundry/concepts/rbac-foundry ↩

SAMPLE

59. https://learn.microsoft.com/en-us/azure/foundry/concepts/rbac-foundry#built-in-roles ↩

60. https://learn.microsoft.com/en-us/ai/playbook/technology-guidance/generative-ai/mlops-in-openai/security/security-plan-llm-application ↩

61. https://learn.microsoft.com/en-us/microsoft-copilot-studio/guidance/responsible-ai#data-privacy-and-security ↩

62. https://learn.microsoft.com/en-us/azure/machine-learning/concept-responsible-ai ↩

SAMPLE

Microsoft 認定 AI 変革リーダー(AB-731) 学習用テキスト(無料サンプル)

版

2026年11月版(AB-731 出題範囲(Skills measured as of January 11, 2027)準拠)

発行

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

お問い合わせ

info@shikaku-doujou.com

Microsoft、Microsoft 365、Microsoft Copilot、Copilot、Copilot Studio、Microsoft Foundry、Azure および Microsoft の製品名・サービス名は、Microsoft Corporation の米国およびその他の国における商標または登録商標です。記載されているその他の会社名・製品名・サービス名は、各社の商標または登録商標です。

本教材は公式シラバスの内容を基に作成したもので、Microsoft Corporation との提携・承認関係はありません。

本文・図表・設問はすべてオリジナルの著作物です。無断での転載・複製・再配布・二次販売はできません。お気づきの点は info@shikaku-doujou.com までご連絡ください。

SAMPLE