

Developing Generative AI Applications on AWS

AWS クラスルームトレーニング

コースの説明

このコースは、ファインチューニングなしで大規模言語モデル (LLM) を活用することに関心のあるソフトウェア開発者に生成 AI を紹介することを目的としています。このコースでは、生成 AI の概要、生成 AI プロジェクトの計画、Amazon Bedrock の開始方法、プロンプトエンジニアリングの基礎、Amazon Bedrock と LangChain を使用した生成 AI アプリケーションを構築するためのアーキテクチャパターンについて説明します。

レベル	実施形式	所要時間
上級	プレゼンテーション、デモンストレーション、グループ演習、ハンズオンラボ	2 日間

コースの目標

このコースでは、以下について学習します。

- 生成 AI について説明し、機械学習とどのように連携するのかを説明する
- 生成 AI の重要性を明確にし、そのリスクと利点について説明する
- 生成 AI のユースケースからビジネス価値を特定する
- 生成 AI の技術的な基本と重要な用語について説明する
- 生成 AI プロジェクトを計画する際のステップを説明する
- 生成 AI を使用する際のリスクと緩和策をいくつか特定する
- Amazon Bedrock がどのように機能するのかを理解する
- Amazon Bedrock の基本的な概念を理解しておく
- Amazon Bedrock のメリットを認識する
- Amazon Bedrock の一般的なユースケースを挙げる
- Amazon Bedrock ソリューションに関連する一般的なアーキテクチャを説明する
- Amazon Bedrock のコスト構造を理解する
- Amazon Bedrock のデモンストレーションを AWS マネジメントコンソールにて実装する
- 基盤モデル(FM) とやり取りする際に、プロンプトエンジニアリングを定義し、一般的なベストプラクティスを適用する
- ゼロショット学習やフューショット学習など、プロンプトテクニックの基本的なタイプを特定する
- ユースケースの必要性に応じて高度なプロンプトテクニックを適用する
- どのプロンプトテクニックが特定のモデルに最も適しているかを特定する

Developing Generative AI Applications on AWS

AWS クラスルームトレーニング

- プロンプトの誤用の可能性を特定する
- 基盤モデル (FM) からの応答に存在する潜在的なバイアスを分析し、そのバイアスを軽減するプロンプトを設計する
- 生成 AI アプリケーションのコンポーネントと基盤モデル (FM) をカスタマイズする方法を特定する
- Amazon Bedrock の基盤モデル、推論パラメータ、主要な Amazon Bedrock API について説明する
- Amazon Bedrock アプリケーションのモニタリング、保護、管理に役立つサービスを特定する
- LangChain を大規模言語モデル (LLM)、プロンプトテンプレート、チェーン、チャットモデル、テキスト埋め込みモデル、ドキュメントローダー、Retriever、Amazon Bedrock 用エージェントと統合する方法を説明する
- 生成 AI アプリケーションを構築するために Amazon Bedrock で実装できるアーキテクチャパターンを説明する
- さまざまな Amazon Bedrock モデル、LangChain、検索拡張生成 (RAG) アプローチを活用するサンプルユースケースを構築してテストする

対象者

このコースは以下のような方を対象としています。

- ファインチューニングなしで大規模な言語モデルを活用することに興味のあるソフトウェア開発者

Developing Generative AI Applications on AWS

AWS クラスルームトレーニング

前提条件

このコースを受講するにあたって、以下の前提条件を満たしておくことをお勧めします。

- AWS Technical Essentials
- Python に関する中級レベルの習熟度

登録

<https://www.aws.training/SessionSearch?pageNumber=1&courseId=145545&trainingProviderId=1&countryName=JP>

コースの概要

1 日目

モジュール 1: 生成 AI の概要 - 可能性を実現する力

- ML の概要
- 生成 AI の基本
- 生成 AI のユースケース
- 生成 AI の実例
- リスクと利点

モジュール 2: 生成 AI プロジェクトの計画

- 生成 AI の基礎
- 生成 AI の実用
- 生成 AI コンテキスト
- 生成 AI プロジェクトを計画におけるステップ
- リスクと緩和策

モジュール 3: Amazon Bedrock 入門

- Amazon Bedrock 入門
- アーキテクチャとユースケース
- Amazon Bedrock の使い方
- デモンストレーション: Bedrock アクセスを設定し、プレイグラウンドを使用する

Developing Generative AI Applications on AWS

AWS クラスルームトレーニング

モジュール 4: プロンプトエンジニアリングの基礎

- 基盤モデルの基本
- プロンプトエンジニアリングの基礎
- 基本的なプロンプトテクニック
- 高度なプロンプトテクニック
- デモンストレーション: 基本的なテキストプロンプトのファインチューニング
- モデル別のプロンプトテクニック
- プロンプトの悪用への対処
- バイアスの軽減
- デモンストレーション: 画像バイアスの軽減

2 日目

モジュール 5: Amazon Bedrock アプリケーションコンポーネント

- アプリケーションとユースケース
- 生成 AI アプリケーションコンポーネントの概要
- 基盤モデルと 基盤モデル(FM) インターフェース
- データセットと埋め込みの操作
- デモンストレーション: 単語の埋め込み
- その他のアプリケーションコンポーネント
- RAG
- モデルのファインチューニング
- 生成 AI アプリケーションのセキュリティ保護
- 生成 AI アプリケーションアーキテクチャ

モジュール 6: Amazon Bedrock 基盤モデル

- Amazon Bedrock 基盤モデル入門
- Amazon Bedrock でモデル推論を実行する
- Amazon Bedrock メソッド
- データ保護と監査可能性
- ラボ: ゼロショットプロンプトを使用したテキスト生成用の Amazon Bedrock モデル の呼び出し

Developing Generative AI Applications on AWS

AWS クラスルームトレーニング

モジュール 7: LangChain を使う

- LLM パフォーマンスの最適化
- AWS と LangChain の統合
- LangChain でモデルを使用する
- プロンプトの作成
- インデックスを使ったドキュメントの構造化
- メモリによるデータの保存と取得
- チェーンを使用してコンポーネントを順序付ける
- LangChain エージェントによる外部リソースの管理

モジュール 8: アーキテクチャパターン

- アーキテクチャパターンの概要
- テキスト要約
- ラボ: Amazon Titan Text Premier を使用して小さなファイルのテキストを要約する
- ラボ: Amazon Titan で長いテキストを要約する
- 質問応答
- ラボ: 質問応答に Amazon Bedrock を使用する
- チャットボット
- ラボ: チャットボットの構築
- コード生成
- ラボ: コード生成に Amazon Bedrock モデルを使用する
- LangChain と Amazon Bedrock のエージェント
- ラボ: Converse API を使用した会話型アプリケーションの構築