

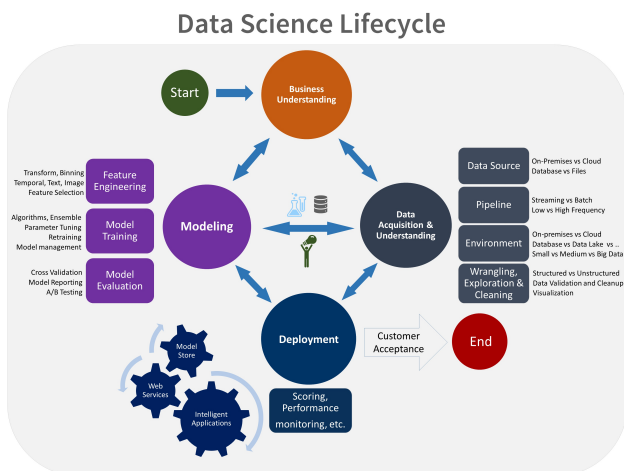
課題 5: クラウドへのデプロイ

背景

機械学習モデルを構築するだけでは十分とは言えません。このモデルを公開し、AdventureWorks のチーム メンバー、開発者、外部の開発者が、顧客向けのアプリケーションやサービスに便利な機能を提供できるようにする必要があります。

この課題では、作成したモデルを REST API としてクラウドで運用化 (Web サービスとしてデプロイ) し、AdventureWorks や外部の開発者が、このエンドポイントを使用した一般ユーザー向けアプリケーションを開発して予測を実施できるようにします。

こちらの概念図では、デプロイフェーズがデータサイエンスのライフサイクルの重要な位置を占めています。



前提条件

- Docker エンジン (Docker for Windows または Docker for Mac) がインストール済みでローカルまたは VM で実行されていること
- Azure CLI および Azure ML CLI (`azure-cli` パッケージおよび `azure-cli-ml` パッケージ)
- `curl` コマンドライン ツールや [Postman \(英語\)](#) などの、モデルのエンドポイントに Request を送るツール
- 課題 3 で保存したモデル

課題

チーム向けのセットアップ環境でチームの代表者が次の手順を実施します。

- 課題 3 または 課題 4 で保存したモデルを、リアルタイム Web サービスとして Azure にデプロイします。

以下のいずれかのツールを使用して、(行列や json の) データ送信および json 応答の受信を行う API としてモデルをデプロイします。詳しくは[参考資料](#)をご覧ください。

- Azure Machine Learning CLI 単体
- Azure Machine Learning Workbench
- Docker で Flask を使用するなどの CLI 以外の手法 (詳しくは推奨事項を参照)

AdventureWorks はできるだけシンプルな API を望んでいるため、json でシリアル化された画像を使用します。

完了条件

- `curl` または Postman で、URL を使用するかシリアル化して画像をクラウドにデプロイした Web サービスに送り、モデルの出力 (登山用品のクラス) を取得する。

参考資料

はじめに

- Azure ML モデルの管理の概要は[こちらのドキュメントを参照](#)
- デプロイメントのガイドは[こちらを参照 \(英語\)](#)

デプロイについての詳細

- Azure ML Workbench と Azure ML CLI のデプロイに関するマイクロソフトのブログは[こちらを参照 \(英語\)](#)
- Azure ML CLI のデプロイのセットアップ方法は[こちらのドキュメントを参照](#)
- CLI 以外のデプロイ方法 (AML の代替手段) については[こちらを参照 \(英語\)](#)

Scoringファイルとスキーマの作成に関する資料

- スキーマ生成の例は[こちらのドキュメントを参照](#)
- サービスへのデータ入力用の CNTK モデルと PANDAS データ型で画像をシリアル化する Scoring ファイルの例は[こちらを参照 \(英語\)](#)
- サービスへのデータ入力用の `scikit-learn` モデルと標準的なデータ型 (json) の Scoring ファイルの例は[こちらを参照 \(英語\)](#)
- 上のリンクの手順に従って `run` メソッドと `init` メソッド、およびスキーマ ファイルを作成した後、[こちらのドキュメント](#)の「モデルを登録する」の手順を実行
 - マニフェスト作成時に `pip` の `requirements` ファイル (`-p` フラグを使用) を使用する場合は、変更は一切必要ありません。

Docker

- Docker のドキュメントは[こちらを参照 \(英語\)](#)

ヒント

- サービスにデータを送る際の入力データ型にはさまざまな種類があり、サービスの呼び出しスキーマを生成するときに指定する必要があります。
- ローカル環境で DSVML とメインの Python を使用して次のコマンドを実行する場合、システムの Python に Azure ML CLI をインストールしておく必要があります (詳細は[こちらのドキュメントを参照](#))
`! sudo pip install -r https://aka.ms/az-ml-ol6n-cli-requirements-file`
- `az ml` コマンドで画像を作成する場合、`conda_dep.yml` やすべてのラベル ファイルなど、必要なファイルはすべて `-d` フラグを付けるようにします。`conda_dep.yml` では、`-c` の使用は避け `-d` フラグを使用します。また、`requirements.txt` を `pip` のインストール形式パッケージで使用する場合は、`-p` フラグを付与します。
- クラスターのデプロイの詳細は[こちらのドキュメントを参照](#)。