Qlik

|ホワイトペーパー

Qlik Analytics および Databricks ベストプラクティスガイド

qlik.com

目次

サマリー 3

はじめに 3

Qlik Analytics アーキテクチャ 5

ユーザーインターフェース 6

Databricks Data Intelligence Platform 7
- 統一 7
- オープン 8
- スケーラブル 8

Qlik および Databricks が提供する高度な統合オプション 8
- Qlik および Databricks 使用オプションの概要 10

Databricks および Qlik メソッドのデシジョンツリーに関する詳細 11
- それでは、どれが貴社に適しているか？ 13

パフォーマンスに関する考慮事項 / テクニック別のベストプラクティス 16
- 増分ロードオプション 16
- 変更データフィードとマージ 18
- オンデマンドオプション 19

Databricks Mosaic Machine Learning との統合 22
- Databricks Mosaic AI 内での基礎モデルの展開 23
- Databricks ML Connector 23
- MLFlow 接続の作成 24
- スクリプトレベルでの MLFlow 接続の使用 26
- チャートレベルでの MLFlow 接続の使用 27

追加リソース 28

サマリー

本ドキュメントは、Qlik Analytics ソリューションと Databricks の統合にあたり留意すべき、次のような側面をカバーすることを目的に作成されました：

- Qlik と Databricks の相乗効果による高度なデータおよび分析パイプライン
- Qlik と Databricks を連動させて展開するための様々なアーキテクチャオプション
- 高度な概念と実用的なアプリケーションの探求
- シナリオに基づき適切なアプローチを選択するためのガイダンス
- Databricks と Qlik プラットフォームの新機能に対応するための継続的なアップデート
- 読者が Qlik と Databricks の両方のテクノロジーに習熟していることを前提としています

はじめに

Qlik Analytics ソリューションは、第 3 世代の分析プラットフォームのベンチマークを確立し、組織内の全担当者がデータに基づいた意思決定を行えるようにします。Qlik 独自の連想エンジン（Associative Engine）を基盤として、セルフサービス分析、インタラクティブなダッシュボード、会話型分析、カスタムおよび組み込み分析、モバイル分析、レポートやアラートなど、データからインサイトに至るまでライフサイクル全体にわって、あらゆるユーザーとユースケースをサポートします。AI を活用したインサイトの提案、自動化、自然言語による対話を通じて、利用する人の直観力を補強し、さらに強化します。Qlik Sense は、SaaS、オンプレミス環境それぞれが持つ利便性、またはその両方によって、比類のないパフォーマンスとガバナンスを提供します。データおよび分析に向けた完全なプラットフォームであり、あらゆるレベルのユーザーが俊敏性と高いパフォーマンスをもってデータを探索できるようにします。Databricks Data Intelligence Platform は、エンタープライズクラスのデータに対する効果的な管理を実現する、クラウドベースの非常にスケーラブルなプラットフォームです。

強力な検索機能と AI 機能を備えた Qlik の連想エンジンと、強力でスケーラブルな Databricks のデータベースエンジン技術を組み合わせることで、Qlik と Databricks は、「思考のスピード」で探索能力を最適化するためのバランスを確保します。

「必要な際に必要なデータを取得する」、「必要な方法でデータを取得する」といった考えを、最先端の 2 つのテクノロジープラットフォームによって融合させることで、エンタープライズ分析、レポート作成、ダッシュボード、データサイエンスなどの機能をビジネスにもたらす独自のソリューションを創出します。

Qlik Analytics アーキテクチャ

Qlik Analytics ソリューションのアーキテクチャは、柔軟かつスケーラブルな形で設計されており、効率的なデータ処理とビジュアライゼーション機能を実現します。中核となる Qlik Analytics ソリューションでは、データがインメモリのエンジンにロードされる連想モデルに従い、異なるデータセット間での迅速かつ動的な関連付けを可能にします。連想モデルによって、ユーザーはデータを直感的に探索および分析することが可能となり、事前に定義された階層や制約に縛られることなく、様々な次元を横断する形でインサイトを発見することができます。

さらに、Qlik Analytics ソリューションではハブベースの構造を採用しており、ユーザーは集中管理されたハブインターフェースを介して共有アプリケーションにアクセスします。この共同作業環境により、チームワークが促進され、データから得られたインサイトを組織全体ですぐに利用できるようになります。このプラットフォームが提供するオープン API と拡張性が、適応性をさらに高め、他のデータソース、カスタムの拡張機能、外部 Web コンテンツとの統合を可能にし、Qlik Analytics ソリューションを最新のデータ分析のための多用途かつ堅牢なソリューションにしています。

お客様は、その嗜好や組織の要件に合わせて、柔軟な形で Qlik Analytics ソリューションを導入することができます。1 つのアプローチには、クライアント管理機能の導入が含まれます。この対応については、お客様自身が分析要件を満たすために必要なインフラストラクチャのインストールと保守の責任を負います。これにより、特定のビジネスニーズに合わせてカスタマイズおよび制御された利用環境を実現することができます。

一方、代替オプションとなるのは、Software-as-a-Service（SaaS）モデルである Qlik Cloud を活用する形態です。このシナリオでは、Qlik Cloud が、セキュリティ、スケーラビリティ、

THE ASSOCIATIVE DIFFERENCE®

リレーショナルデータベースやクエリは、1980 年代にトランザクションシステム用に設計されたものであり、当然、最新の分析に向けられたものではありません。クエリベースのツールは、データを取りこぼし、ユーザーを制限のある画一的な探索に縛り付けてしまうため、結果的に見落としや機会損失を招きます。

Qlik Sense は、独自の Qlik Associative Engine 上で動作し、あらゆるスキルレベルのユーザーが制限なく自由にデータを探索できるようにします。Qlik Associative Engine は、どんなデータでも取りこぼすことなく、その大小を問わずデータを無制限に組み合わせて活用します。インタラクティブな選択と検索を通じて、これまでにない自由な探索を実現し、すべての分析内容を瞬時に再計算し、緑（選択済み）、白（関連する）、灰色（無関係）のカラー表示で、ユーザーにその関連性を示します。Qlik Associative Engine は、すべてのビジュアライゼーションをコンテキスト内でまとめて保持し、“関連する値”と“無関係な値”の両方を分析機能内に保持することにより、ユーザーがクエリベースのツールでは見逃してしまうような、隠れたインサイトを発見できるようにします。

Qlik Associative Engine は、大量のデータを大規模かつ動的に、計算および関連付けるために設計されており、多くのユーザーに対応できる高いスケーラビリティを備えています。この独自のテクノロジーは、30 年以上にわたる革新と投資によって得られた Qlik のアドバンテージと言えます。

パフォーマンスの最適化を含む分析サービスのあらゆる面に関する責任を負うことになります。これにより、合理化された手間のかからないソリューションが提供されるため、組織は基盤となるインフラストラクチャの管理に煩わされることなく、分析に向けたインサイトの活用に集中できるようになります。複雑なインフラストラクチャのメンテナンスが不要で、スケーラブルかつセキュアな分析ソリューションを求めるユーザーにとって、Qlik Cloud のアプローチは特に有益なものとなります。

ユーザーインターフェース

Qlik Sense Enterprise SaaS 環境へのアクセスは、特別なインストールを必要とせず、どんなシステムからでも利用が可能な、ゼロフットプリントの Web ブラウザーインターフェース（Qlik Sense Hub と呼ばれます）を介して行われます。Qlik Sense Web ブラウザーインターフェースにより、開発、ドラッグ & ドロップによるコンテンツの作成、その利用など、あらゆる対応が可能となります。

Qlik Sense は、ラップトップ、デスクトップ、タブレット、スマートフォンのブラウザーなど、利用デバイスに合わせて適切なレイアウトや情報を含むビジュアライゼーションを自動的化しサイズ調整を行う、レスポンシブデザインを採用しています。Qlik Sense は、HTML5、CSS3、JavaScript®、Web ソケットなど、現在の業界標準に準拠して構築されているため、あらゆるデバイスでアプリを構築して使用することができます。

Web ベースのインターフェースに加え、Qlik Sense では、Slack や MS Teams などの主要なチャットプラットフォームと統合された会話分析機能や、ユーザーがデータをサブスクライブして、データに対する重要な変更通知を受け取ることができるデータアラート機能をサポートしています。

Qlik Mobile は、Qlik の分析プラットフォームが提供する機能を外出先のユーザーにも拡張し、様々なモバイルデバイスに対してシームレスでレスポンスに優れたエクスペリエンスを提供します。モバイルアプリケーションは、ユーザーがスマートフォンやタブレットで Qlik Sense 分析機能にアクセスして、探索や操作を行えるように設計されています。Qlik Mobile を使用することで、意思決定者は常に重要なインサイトやデータ駆動のビジュアライゼーション機能にアクセスし、いつでもどこでも情報に基づいた意思決定を行うことが可能になります。

このアプリケーションは、ユーザーフレンドリーなインターフェースを保証し、Qlik Sense のインタラクティブで連想的な特性を維持しながら、小さな利用画面向けに複雑なビジュアライゼーションの

表示内容を最適化します。Qlik Mobile はレスポンシブデザインの原則に従い、様々な画面サイズや方向に適応して全体的なユーザーエクスペリエンスを向上させます。オフィス内にいるときも、会議中でも、さらに移動中の場合でも、ユーザーは Qlik の連想分析の全機能を活用し、データ探索と意思決定に向けてモバイルファーストのアプローチを促進することができます。

データ統合機能、カタログ作成、Qlik Sense の拡張を含む Qlik プラットフォーム全体へのクイックリファレンスは、Databricks 用の統合スイートの強力な機能を示しています。

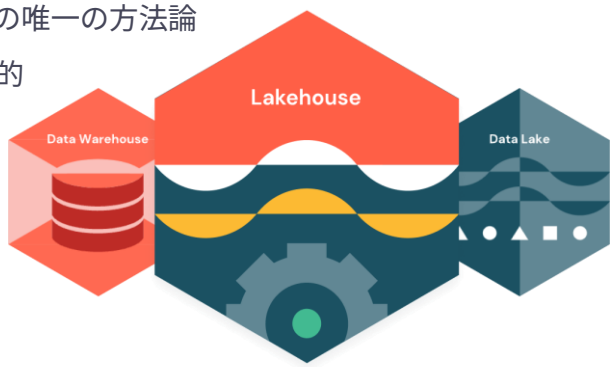
Databricks Data Intelligence Platform

Databricks Data Intelligence Platformの基盤は、革新的な形でデータレイクとデータウェアハウスを組み合わせたレイクハウスアーキテクチャ内に存在します。この革新的なアプローチは、コストを最小限に抑え、データおよび AI に関わるゴールの迅速な達成を目的としています。

オープンソースの原則を取り入れ、オープンスタンダードに準拠したレイクハウスアーキテクチャは、データおよび AI 分野を複雑にすることが多かった従来の障壁を取り除くことで、データインフラストラクチャを簡素化します。これにより、データリソースを管理および活用するための、より一貫性のある効率的な環境が提供されます。

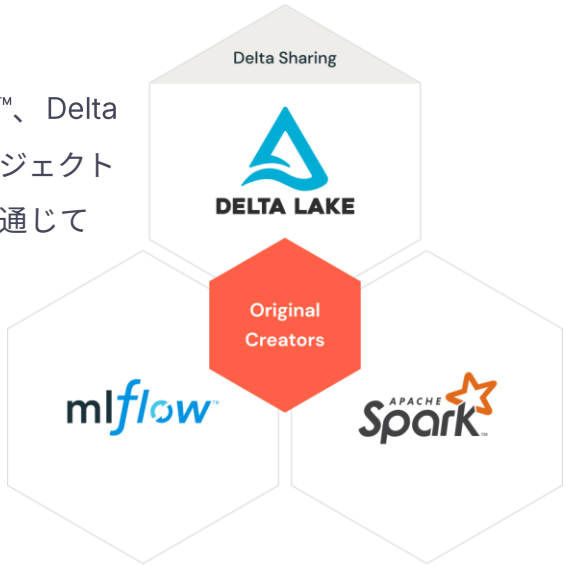
統一

統合、ストレージ、処理、ガバナンス、共有、分析、AI を含む統一アーキテクチャです。構造化データおよび非構造化データの両方を処理するための唯一の方法論になります。データの系統と出所を最初から最後まで包括的に把握できる視点を持ちます。Python や SQL、notebooks や IDE、バッチプロセスとストリーミング処理を、すべての主要なクラウドプロバイダーに対応させることができる一貫性のあるツールキットです。



オープン

Databricks フレームワーク内では、データの制御が一貫して維持され、独自のフォーマットや閉鎖的なエコシステムからの独立性が確保されます。



レイクハウスアーキテクチャの基盤は、Apache Spark™、Delta Lake、MLflow など、広く採用されているオープンソースプロジェクトに依存しています。Databricks パートナーネットワークを通じてグローバルなサポートを受けることができます。さらに、Delta Sharing機能は、レイクハウスから任意のコンピューティングプラットフォームへのリアルタイムデータを安全に共有するためのオープンなソリューションを提供します。本機能は、データレプリケーションや複雑なデータ抽出、変換、ロード（ETL）プロセスを必要とせずに実施されます。

スケーラブル

パフォーマンスおよびストレージの自動最適化機能は、データプラットフォームの中で総所有コスト（TCO）が最低になるように細心の注意を払って設計されており、同時にデータウェアハウジングやMosaic人工知能（AI）のユースケースで世界記録に匹敵するパフォーマンスを達成しています。この対応は、大規模言語モデル（LLM）などの生成技術の適用にまでおよびます。

Databricks は、スタートアップ企業からグローバル企業に至るまで、その規模に関わりなく、該当企業の運用要件に効果的に対処できるように設計されています。

Qlik および Databricks が提供する高度な統合オプション

Qlik および Databricks は、リソースを最大限に活用し、パフォーマンスを維持し、組織全体のビジネス要件を満たすために、特定のシナリオに適した様々な統合機能を提供します。同時使用ユーザー数、データの更新頻度、パフォーマンス、ユーザーエクスペリエンス、および総コストによって、どの統合オプションを使用すべきかが決定されます。 Qlik では、Databricks からデータをロードして使用するためのいくつかの手法を提供しています。次に示す表は、それぞれの統合構成について説明しており、リファレンスガイドとして自由にご利用いただけます。

統合オプション	手法	ユースケース
インメモリ	完全/増分ロード（QVD）	最も一般的で最大のパフォーマンスを発揮。QVD を利用した増分更新および分析エンジン用に最適化されたバッチリロードの実施
インメモリ+ Databricks	完全/増分ロード（Change Data Feed）	Databricks 向けに最適化されたリロード – データについて現在および過去の状態を比較する必要がある場合や、増分リロードの場合に特に有効
オンデマンド	ODAG	大量データへのアクセスに向けた、詳細情報への階層化されたドリルダウン
オンデマンド	ダイナミックビュー	既存のアプリ内で、ビジュアライゼーションオブジェクトとしてオンデマンドで詳細情報をサポート

Qlik および Databricks 使用オプションの概要

Qlik Sense へのデータロード方法を指定するインメモリオプション：

1. スケジュールに従って毎回、完全リロードを行う：

- データ量によってはロード時間が長くなる可能性があります。
- データの揮発性が高い場合、または変更回数が多い場合に推奨します。

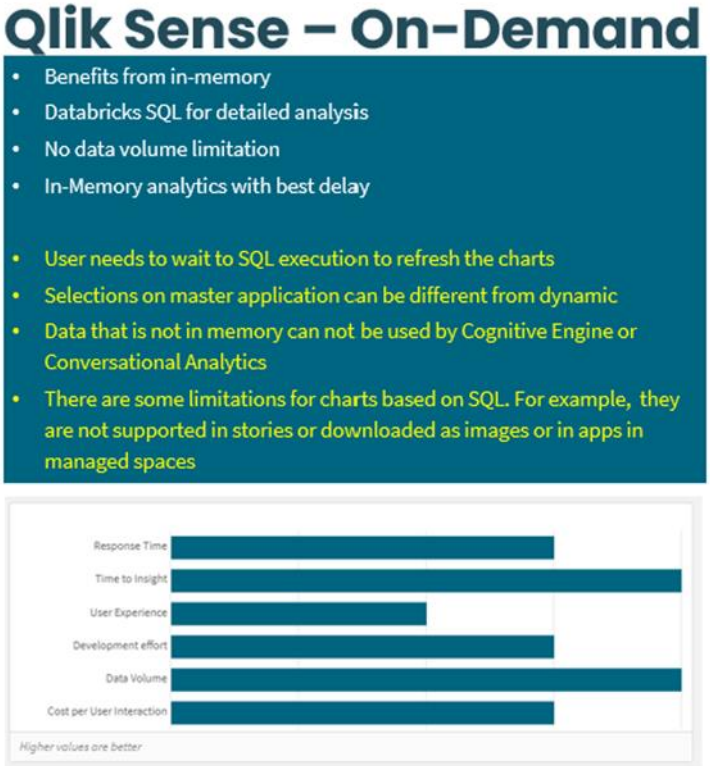
2. 新しいデータのみ増分ロードを行う：

- 事前定義されたスケジュールに従ってデルタリロードを開始します。本機能については、Qlik 開発者が、抽出、ロード、変換（ELT）のスクリプト内で構成することができます。これにより、Qlik がデータ内で排他的に更新作業を行えるようになります。この方法を採用することで、Qlik エンジンおよびビジュアライゼーションレイヤー内でのデータの準リアルタイム更新が可能となります。

オンデマンドアプリ：

1. On Demand App Generation（ODAG）

- オンデマンドの詳細データ：本アプローチは、一般的に大規模なデータセットを含むシナリオで使用されており、同シナリオでは Qlik アプリがサマリーデータを格納するよう設計されています。Qlik 詳細アプリはテンプレートとして機能し、サマリーアプリからパラメーターを受け取り、動的に Databricks へライブデータをクエリします。ユーザーは、自らの選択に基づいて関連するデータスライスにアクセスすることができます。この手法は、サマリーと詳細の分析を実行する場合に効果的となります。



2. **ダイナミックビュー**

ライブデータのビジュアライゼーション：Qlik アプリにおいて、ライブまたはニアリアルタイムのデータを必要とするシナリオで採用されます。ODAG フレームワークを使用することで、ユーザーの操作に基づいてライブデータの更新機能がトリガーされ、ユーザーが選択内容の更新を選択すると、リアルタイムで Databricks からデータが再ロードされます。

3. **Direct Query アプリケーション**

Direct Query は Qlik Cloud Analytics の 1 機能であり、ユーザーがビジュアライゼーションを行い、動的フィルターを適用する際、分析アプリがクラウドデータベースに対してリアルタイムで直接 SQL クエリを生成できるようにします。これによりデータの事前ロードが不要となり、データ量の制限も発生せず、パフォーマンスは基盤となるクラウドデータベースによってのみ決定されるようになります。

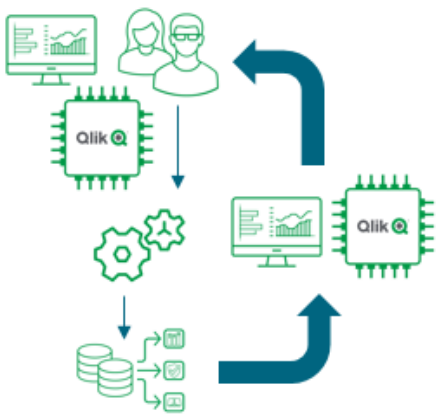
4. 機械学習機能の統合

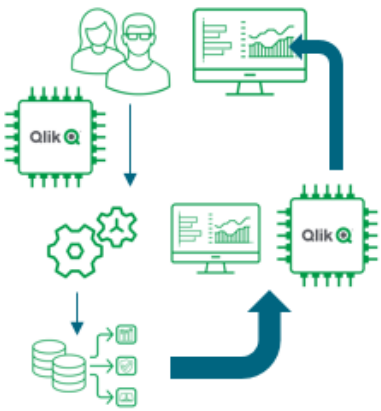
Qlik Cloud Analytics は、Qlik 連想エンジンとサードパーティの機械学習モデルとの間の接続を実現する分析接続機能を提供します。この機能には、REST ベースの API エンドポイントを公開する接続の生成が含まれ、Qlik と外部機械学習モデル間のデータのシームレスな交換を可能にします。分析接続は、Qlik Cloud Analytics 環境内で機械学習を活用するためのブリッジ機能を提供します。

Databricks および Qlik メソッドのデシジョンツリーに関する詳細

多くの組織には、確立されたデータパイプラインと、データやインサイトへのアクセスを求めるユーザーベースが存在します。以下のセクションは、Databricks と連携して展開を行う際に Qlik Sense で利用できる様々な統合オプションの概要をご紹介し、特定の要件に基づいた意思決定を支援することを目的としています。

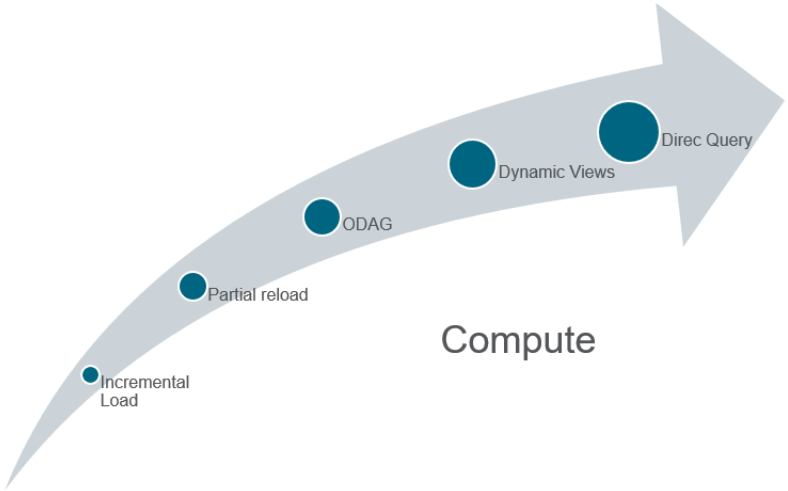
最適な統合オプションを決定する際には、同時使用ユーザーの総数、システムパフォーマンスのしきい値、コスト、特定のビジネス要件などの要素を考慮する必要があります。以降、お話を進めるにあたり、様々なビジネスユースケースで使用される潜在的なアプリケーションを詳しく掘り下げる前に、まず各 Qlik 統合オプションの概要について理解を深めましょう。

増分ロード（スケジュールされたデータロード） キャッシュ型インメモリ	増分ロードは、Qlik が必要なテーブルの最新データをロードし、変更内容を既存の QVD（以前にロードされたデータ/キャッシュ）とマージして、それらの QVD/キャッシュのリライトを行うようスケジュールされたプロセスです。これにより、すべてのアプリケーションやユーザーが該当テーブルの更新情報にアクセスできるようになります。このアプローチは、大規模なデータソースを扱う際に特に有効であり、最初の一括ロード後に事前定義されたスケジュールに従って変更データのみをロードすることで、データの鮮度と処理速度のバランスを保ちます。
部分リロード キャッシュ型インメモリ ＋ インメモリとして追加された新規データ (オンデマンドまたはスケジュール済み)	完全ロードまたは増分ロードでは、既存のデータモデル内のすべてのテーブルが削除され、ロードスクリプトが実行されます。一方、部分リロードの場合には、データモデル内のすべてのテーブルが保持され、Add、Merge、または Replace プレフィックスを含む Load ステートメントおよび Select ステートメントのみが実行されます。このアプローチでは、一部のテーブルの変更されたデータが連想エンジンにロードされ、データの鮮度と処理速度のバランスが確保されます。特にデータソースが変更（挿入、更新、削除）に関する詳細を提供している場合には、ユーザーがアクションを開始して最新の値を表示する際の信頼性が高まります。
On Demand App Generation - ODAG （ライブデータロード） キャッシュ型インメモリ 選択項目 ＋ ユーザーが問合せた際の 2 番目のアプリケーションでの選択に基づくライブ データ	On Demand App Generation は、エンドユーザーが事前定義されたテンプレートアプリケーションに選択内容を提供するためのプロセスです。Qlik はテンプレートをコピーし、特定のコホートに向けデータをロードして、新しいアプリケーションとしてエンドユーザーに提示します。呼び出し元のアプリケーションは、集計されたデータを完全にメモリ内に保持し、迅速なアクションを可能にすると共に、生成されたアプリケーションは、必要な場合にのみライブデータを取得します。 

ダイナミックビュー （ライブデータロード） キャッシュ型インメモリ 選択項目 ＋ ユーザーが問合わせた際、選択内容に基づいてライブデータからチャートを表示	ダイナミックビューは ODAG に類似したプロセスですが、エンドユーザーに対して新しいアプリケーションを表示する代わりに、現在使用しているアプリケーション内でテンプレートアプリケーションから選択されたチャートを表示します。ユーザーが選択内容を変更した場合、ダイナミックビューのリフレッシュを要求することができます。  このオプションは、ODAG の機能を活用すると共に、ユーザーが選択したアプリケーションのコンテキスト内でデータの「ライブ」コホート分をレンダリングします。
Direct Query	データボリュームが膨大な場合、または頻繁に変更が行われ、Qlik Engine にデータをロードすることが困難になる場合には、このアプローチを使用することができます。
分析接続	Data Intelligence Platform に展開された ML モデルと統合する必要がある場合に使用します。このオプションは、スクリプトレベルで（スケジュール済みのリロードとして）統合することも、チャートレベルでのビジュアライゼーション中に（リアルタイムで）統合することもできます。

それでは、どれが貴社に適しているか？

適切な問題解決策を選ぶための前提として、各ソリューションがどれだけのコンピューティングリソースを消費し、その結果としてコストにどのような影響を与えるかという面に焦点を当て、非常にハイレベルなガイダンスを提供します。多くの場合、「ライブデータが欲しい」という要求と「1分前なら問題ない」という要求の違いは、実装コストに起因する可能性があります。他のユースケースでは、問題解決に関連するコンピューティングリソースやコストに関わりなく、ライブデータは必須となります。次に示すチャートには意図的に目盛を入れていません。これは、シンプルに各テクノロジー関連のコンピューティング要件が増大するという概念を表現するためのものだからです。



「各ソリューションのコンピューティング量やコストが増加するのはなぜか?」という質問の答えについて考えてみます。増分ロードを実行するということは、Qlik がスケジュールされたリロードごとに 1 回だけ、ソースからの変更を要求することを意味します。On Demand Application Generation（ODAG）アプリケーションは、仮に何年も変更されていない場合でも、定義されたコホートのすべてのデータをロードします。 ダイナミックビューは、ODAG アプリケーションと同様のデータロードを必要とし、エンドユーザーが選択内容との同期を維持するために必要な分だけリフレッシュできるため、より多くのコンピューティングパワーが必要になります。ユーザビリティの観点から言えば、ダイナミックビューについては、ユーザーがアプリケーションを唯一のデータソースとして認識するため、ODAG よりも頻繁に使用される傾向があります。ODAG で「新しいアプリの生成」ボタンが押された場合とは異なり、ダイナミックビューでは、SQL クエリの実行を常にユーザーが意識しているわけではありません。さらに Direct Query を使用すると、ユーザーが行うあらゆる選択に対して、すべてのクエリが Databricks Data Intelligence プラットフォームへプッシュダウンされます。

	ただし、これはあくまで一般的なガイドにすぎないことを覚えておいてください。特定のビジネスユースケースでは、少数のエンドユーザーのみが使用するアプリケーション内での集計に向け、1 つの SQL しか必要とされない場合もあります。このため、次のポイントに移ります：ビジネス主導のユースケースは複雑な場合が多く、様々な質問に対する回答が必要となるためです。

次のシナリオでは、各アーキテクチャソリューションに対応する仮想のユースケースをご紹介します。

シナリオ 1 – ODAG：詳細レポート/分析に向けたドリルダウン

主要な指標のサマリーを行うアプリケーションから始まり、ユーザーが選択した基準が、オンデマンドで生成される 2 番目のアプリケーションへ渡されます。以下は、ODAG を使用する際のお客様の例です：

某人事部門には、全社で 1,290 名の人事管理者、マネージャー、従業員が使用し、頻繁に選択を行う既存アプリケーションがあります。現状このアプリケーションは、1 時間ごとの増分ロードというスケジュールで運用されています。同部門の担当者達は、潜在的な時間外労働の問題を確認するためにデータのリロードを待つのではなく、リアルタイムデータを使用したいと考えています。そのデータの詳細は、潜在的な時間外労働の問題を示す特定のフォーマットでのコンテンツの既知のサブセットであるため、ODAG では、簡単に利用できる Qlik アプリケーション内で一時的なリロード機能を提供します。

シナリオ 2 – ダイナミックビュー：コンテキスト内トランザクションの詳細

このシナリオは特定のトランザクションの詳細な内容を、元の Qlik Sense アプリケーションのコンテキスト内のチャートとして表示する必要がある場合に最適となります。以下は、ダイナミックビューを使用する際のお客様の例です：

某顧客企業はトランザクションシステムからの数千億件のレコードを Databricks に格納していますが、これは単一の Qlik アプリケーションでは処理できないデータ量です。これらのデータは、KPI（主要業績評価指標）およびその他の関連データの集計値と共にスケジュールに従ってリロードされますが、トランザクション自体はリロードされません。ダイナミックビューは、特定のディメンションのコホートに関連したトランザクションを取得するために使用され、しきい値（例えば10万行以下）で制限したレコードを分析するために、要求に応じて任意のチャート内に表示します。アプリケーションのユーザーは、選択したコホートのコンテキスト内で詳細な内容を確認できる必要があります。

シナリオ 3 – 部分的なリロード/マージ：決算書/財務諸表

月末に帳簿を締める財務リーダーは、総勘定元帳の最新の詳細情報にアクセスする必要があります。Qlik アプリケーションには、SQL では簡単に複製できない非常に複雑な計算、階層化、変換機能が含まれています。約 25 名の企業内会計士および上級幹部だけが、このアプリケーションを使用しています。決算のサイクルを通じてデータが急激に変化するため、ユーザーは常に会社の現状を把握する必要があります。エンドユーザーは、非常に高度なインタラクティブ性をもって問題を見つけ出すために、データを分析する必要があります。

パフォーマンスに関する考慮事項 / テクニック別のベストプラクティス

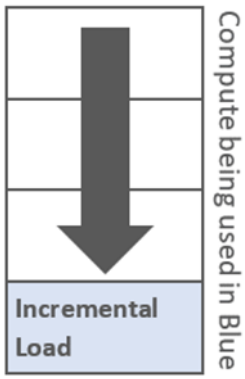
前述のとおり、Qlik と Databricks は連携する形で、Qlik の分析エンジンを使用したメモリへの様々なレベルでのデータのロードや、ニアリアルタイムでのデータのロードなど、幅広いデータ取り込み手段を提供します。それぞれの手段には明確な強みや優位性がありますが、一方でユースケースが限定されるものもあります。

Databricks および Qlik間のキャッシングプロセスを理解することで、各システムを最適化し、分析ニーズを効果的にサポートする方法を明確化することが可能となります。

増分ロードオプション

次のセクションでは、ユースケースに応じて Databricks データを活用するための上記の戦略について詳しく説明します。

1. **QVD を使用した増分ロード**



この手法は、変更されたデータや新しいデータのみを効率的にロードするために、Qlik 環境で広く使用されています。QVD は、バイナリ形式で最適化されたデータフォーマットであり、スナップショットの取得、データ構造の再利用、そして増分ロードを可能にします。このプロセスには以下が含まれます：

1. 大規模なテーブルを特定
2. 最初にすべてのデータをロードし、QVD ファイルとして保存
3. 後続のロードでは、特定の列に基づいて新しいレコードを識別
4. WHERE 句を使用して Databricks から得られた新たなデータを識別済みの変数とマージ
5. 既存の QVD を上書き

この手法により、データのリロードが最適化され、パフォーマンスが向上し、スナップショットベースのデータ分析が可能になります。詳細な説明については、Qlik コミュニティの投稿をご覧ください： https://community.qlik.com/t5/Qlik-Design-Blog/Qlik-Sense-Incremental-Load-using-Merge/ba-p/1944225

2. **部分ロードを使用した増分ロード**

完全なリロードは、現在のデータモデル内のすべてのテーブルの削除から開始され、その後、ロードスクリプトが実行されます。これと対照的に、部分リロードでは、データモデル内のすべてのテーブルが保持され、Add、Merge、または Replace プレフィックスを含む Load ステートメントと Select ステートメントのみが実行されます。

部分ロードは、既存のテーブルに追加する新しいデータはあるが、残りのテーブルについてはロードしたくない場合に役立ちます。これは、リアルタイムで多くの変更の可能性がある 1 つのテーブルと、データモデルの残りの部分に多くのソースから得られた大規模なデータセットが存在する場合にも役立ちます。

ステートメント	完全リロード	部分リロード
ロード ...	ステートメントが実行されます	ステートメントは実行されません
Add/Replace/Merge Load ...	ステートメントが実行されます	ステートメントが実行されます
Add/Replace/Merge Only Load ...	ステートメントは実行されません	ステートメントが実行されます

使用例：

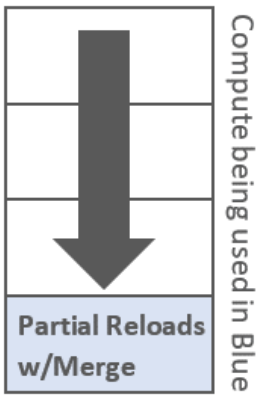
ADD LOAD * from Databricks.FactTable where createdatetime > $(lastreloaddatetime);

上記の例は、リロード時に「Partial」オプションが設定されている場合、および、部分フラグが設定されていない場合に実行されます。

ADD ONLY LOAD * from Databricks.FactTable where createdatetime > $(lastreloaddatetime);

上記の例は、Partial フラグが設定されている場合にのみ実行され、リロードで Partial が false と設定されている場合には実行されません。

変更データフィードとマージ



従来、Qlik 開発者は、最終変更タイムスタンプ（Last Modified Timestamp）フィールドを利用して、変更された値のみをロードできるようにしていました。しかし、データテーブルに最終変更タイムスタンプが存在しない場合が多いため、完全リロードが唯一の選択肢となっていました。

Databricks は変更データフィードを提供します。これは、クエリを使って Change Data Capture 情報を簡単に取得するための機能です。この機能を（テーブルごとに）有効化すると、バックグラウンドでテーブルにメタデータカラムが追加され、通常のSQL 選択で「where」句を使用して、直近の実行以降の変更内容のみを取得できるようになります。

```
%sql
-- view the changes
SELECT * FROM table_changes('silverTable', 2, 5) order by _commit_timestamp
```

	Country	NumVaccinated	AvailableDoses	_change_type	_commit_version	_commit_timest
1	Australia	100	3000	insert	2	2021-04-14T20:2
2	USA	10000	20000	update_preimage	3	2021-04-14T20:2
3	USA	11000	20000	update_postimage	3	2021-04-14T20:2
4	UK	7000	10000	delete	4	2021-04-14T20:2

Showing all 4 rows.

更新されたレコードの新しい値のみが対象であるため、_change_type = ‘update_preimage’ レコードは無視することができます。

Databricks から得られたこの CDC キャプチャ情報を Qlik Merge 関数と組み合わせることで、増分ロードを処理するためのハイパフォーマンスな手段が提供されます。Merge 関数は、メモリ内のテーブルへ変更内容を自動的に「マージ」するための簡単な手段を提供します。対応は関数にいくつかのパラメーターとテーブルに対する Databricks Changes クエリの結果を渡すだけです。

テーブルで Change Data Feed が有効になっている SQL 例

```
[silvertable]: // full load
SELECT
   Country,
    NumVaccinated,
    AvailableDoses
FROM
   `default`.silvertable;
```

```
If IsPartialReload() then
  //Sets a default starting point in time if this is the very first time
  Let Last_commit = Alt(chr(32)&Timestamp(Last_commit,'YYYY-MM-DD hh:mm:ss.ff')&chr(32),1);
  // Incremental load
  Merge only (_commit_timestamp, Last_commit) on Country Concatenate(silvertable)
  SELECT
     left(_change_type,1) as Operation, // We need only the first character
     Country,
     NumVaccinated,
     AvailableDoses,
     _commit_timestamp
  FROM table_changes('silverTable', $(Last_commit))
  WHERE _change_type !='update_preimage'
  ORDER BY _commit_timestamp;

  Drop field _commit_timestamp from [silvertable];//Optional
End if
```

この機能（関数）によって以下が実現されます:

1. 従来の増分ロードアプリケーションが高速化されます。
2. 最終変更タイムスタンプカラムのないテーブルで完全リロードを実行する可能性がある場合でも、増分ロードを処理するために使用することができます。
3. 部分リロードを使用することで、エンドユーザーは最新かつ最大の変更内容を迅速にメモリ内に取り込み、分析できるようになります。

Databricks 変更データフィード（Change Data Feed）の詳細については、https://docs.databricks.com/delta/delta-change-data-feed.html をご覧ください。

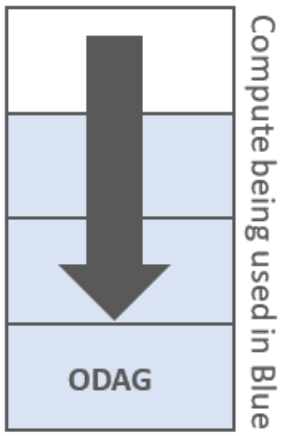
Qlik Merge 関数の詳細については、https://help.qlik.com/en-US/sense/May2022/Subsystems/Hub/Content/Sense_Hub/Scripting/ScriptPrefixes/Merge.htm をご覧ください。

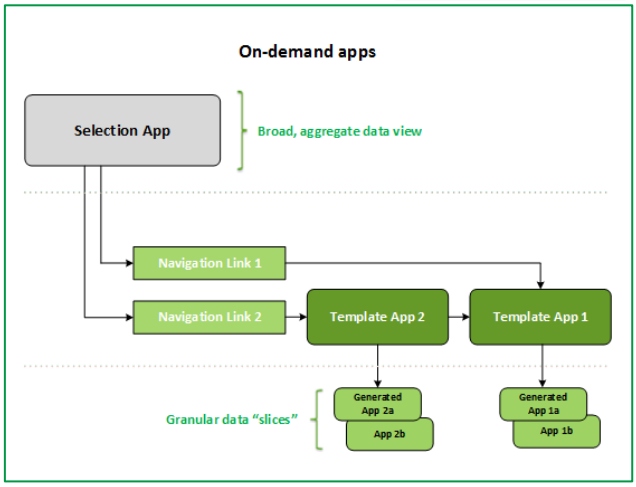
オンデマンドオプション

次のセクションでは、Databricks から得られたリアルタイム/ニアタイムデータの活用戦略について詳しく説明します。

On-Demand App Generation（ODAG）

多くの場合 ODAG と呼ばれるこの手法には、分析に向けて特定のデータのサブセットを取り込む機能が含まれています。特に、すべてのデータを Qlik アプリに読み込むことが現実的ではない、大量のデータが対象となるシナリオで有効となります。このアプローチは、一般的にファクトテーブルの行数が 5億～8億行を超える場合に採用されます（ただし、様々な要因によって異なる可能性がある概算値です）。例えば、「ショッピングカートアプリ」の場合、最初にユーザーはハイレベルの集計データを探索し、その後選択を行いながら分析内容を絞り込んでいきます。希望するパラメーターで管理できるデータセットを確認したら、チャートオブジェクトやレイアウトが事前定義された「分析アプリ」へ移動するか、必要に応じてデータを空のシートにロードすることができます。



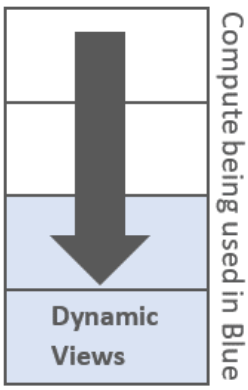


オンデマンドアプリの作成と管理に関する公式のヘルプドキュメントについては https://help.qlik.com/en-US/sense/May2022/Subsystems/Hub/Content/Sense_Hub/DataSource/Manage-big-データ.htm をご覧ください。

ダイナミックビュー

ダイナミックビューを使用することで、ユーザーは調査したい分析ソースをアクティブに管理し、ビジュアライゼーションによってデータが更新されるタイミングをコントロールすることができます。

これらのビューは、ユーザーが大規模データセットの特定のサブセットをクエリし、その選択内容に応じて動的に更新されるチャート上に表示されます。本機能により、ビジュアライゼーションの内容が常に最新の状態に保たれるため、大量のデータや急速に変化するデータが含まれるシナリオに向け最適なものとなります。



ダイナミックビューは、ベースとなるアプリを他のアプリへリンクする機能を提供しており、後者のマスタービジュアライゼーションを前者へ組み込むことができるようにします。これによりアプリ作成者は、ベースアプリ内のダイナミックビューの数に制限を設けることなく、異なるアプリ内でマスタービジュアライゼーションを動的チャートとして活用できるようになります。
ダイナミックビューは、次の 3 つの主要コンポーネントで構成されています。

1. **ダイナミックビュー：** ベースアプリに統合された機能であり、テンプレートアプリに接続することで、アプリ作成者がテンプレートアプリのマスタービジュアライゼーションをベースアプリへ組み込めるようにします。
2. **ダイナミックビューテンプレートアプリ：** クラウドデータベースなどのデータソースへの接続機能を含む Qlik Sense アプリであり、ダイナミックビューのソースとして機能します。
3. **ダイナミックチャート：** ベースアプリに追加できるダイナミックビューテンプレートアプリ内のマスタービジュアライゼーションです。ユーザーはこれらのダイナミックチャートを手動で柔軟に更新することができます。

ダイナミックビューによって、テンプレートアプリとベースアプリの間で、異なるデータセットを柔軟に操作できるようになります。例えば、ベースアプリが顧客の購入に焦点を当てている場合、気象データを含むテンプレートアプリからダイナミックビューを追加することで、潜在的な相関関係を探索できるようになります。

また、テンプレートアプリからクエリされたデータがベースアプリの選択内容と一致するように、テンプレートアプリのスクリプト内でバインディング式を使用します。これにより、ベースアプリでの選択内容に基づいて、ダイナミックビューがテンプレートアプリのデータソースからデータのサブセットを選択的にクエリできるようになります。例えば、ベースアプリの SalesDate フィールドを、テンプレートアプリの DailyTemperatureReadingDate フィールドにバインドすることができます。

ダイナミックビューの作成と管理に関する詳細なガイダンスについては、公式ヘルプドキュメントをご覧ください。

Direct Query アプリケーション

Direct Query によってユーザーは、個々のニーズに合致したデータアクセスの方法について、より多くの選択肢を手にすることができます。Direct Query を通じてデータにアクセスすることで、ユーザーは基礎となるデータソースにデータを保持したまま利用することができます。これにより、ユーザーはデータ操作を高速化できますが、一方で Qlik Cloud アプリが提供するメモリ内の柔軟性の一部を犠牲にすることになります。

一般的に、データについては可能な限り Qlik Cloud へインポートすることが推奨されています。インメモリ Qlik Cloud アプリを使用することで、より適切にエクスペリエンスをカスタマイズし、データを最大限に活用できるようになります。ただし、データをインポートしても目標を達成できない状況に陥った場合には、Direct Query が解決策になる可能性があります。Direct Query アプリは提供する合理的な機能により、新しいユーザーが完全に機能する高速なインメモリアプリ作成に向けた最初の一歩を踏み出す際にも有効となります。

次のような場合には、Direct Query の使用をご検討ください。

- Qlik Engine のメモリ内に収まらないビッグデータソースの処理
- 新しいデータベースおよびテーブルの探索
- 例えば、ほぼリアルタイムなデータとして過去 1 時間に届いた注文件数の確認
- 実稼働前の初期段階でのダッシュボードのプロトタイプ開発
- ODAG アプリを介したデータスライスの Qlik Sense エンジンへの抽出

Direct Query アプリの作成時に利用できる機能については、Qlik ヘルプの「サポート機能（Supported Capabilities）」セクションをご覧ください。

Databricks Mosaic Machine Learning との統合

データインテリジェンスプラットフォームは、統一プラットフォームとして機械学習機能を提供します。同プラットフォームでは、Llama、TensorFlow、PyTorch、scikit-learn などの人気の高いフレームワークを使用して、機械学習モデルの構築、トレーニング、展開を大規模に実施できるほか、AutoML 機能を活用して、Qlik Analytics に分類や予測機能を追加することも可能です。

Qlik Cloud Analytics を Databricks Mosaic Machine Learning と統合することで、両プラットフォームの長所を組み合わせた強力なデータ分析ソリューションが実現されます。Qlik Cloud Analytics は、Databricks のテーブルや Delta Lake などを含む様々なソースから得られたデータへのシームレスなアクセスやビジュアライゼーションを可能にします。Databricks Mosaic Machine Learning は、大規模データセットにおける複雑なデータ処理やモデリング作業に長けており、実行結果はシームレスな形で Qlik Sense へ提供され、詳細な分析やプレゼンテーションが可能となります。

この章では、Qlik-Databricks Mosaic ML コネクタを使用した、Databricks Mosaic AI 内での Qlik Cloud Analytics と Mosaic AI Foundational モデルの統合プロセスについて説明します。また、実際の

ユースケースを紹介すると共に、様々な業界やシナリオにおけるこの統合対応のメリットについてハイライトします。

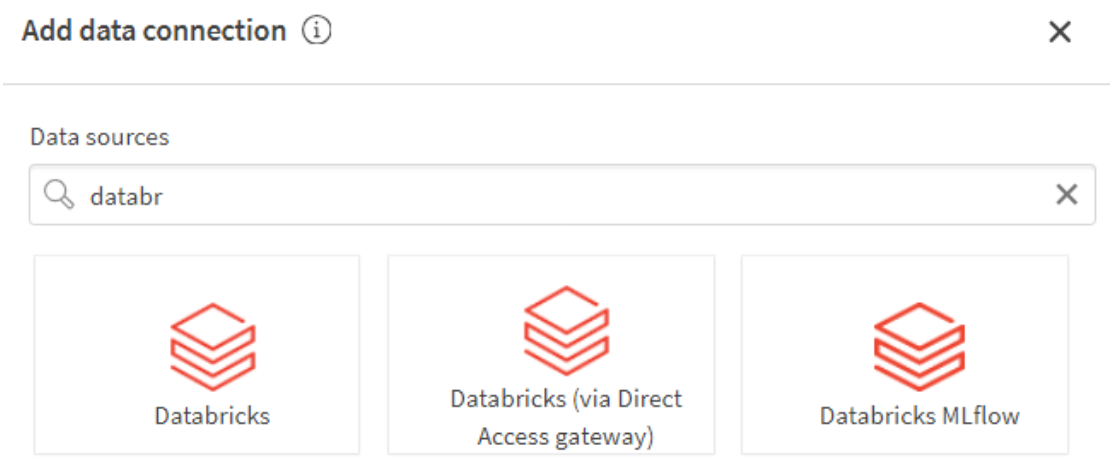
Databricks Mosaic AI 内での基礎モデルの展開

Databricks Mosaic AI 内に Databricks ML モデルを展開する場合、データサイエンティストは特定のニーズに基づいて意思決定を行います。レイテンシーに関する厳格な制約のない、多数の予測を必要とするユースケースの場合、一般的なアプローチにはバッチ推論が含まれます。この対応では、モデルに対する大量データの提供、予測の生成、そして Lakehouse 上のテーブルへのそれらの記録が必要になります。Qlik Analytics は、モデルによってデータが格納されたテーブルから結果を読み取ることができます。低レイテンシーでの予測が求められるシナリオ、例えばアプリケーション内でのユーザーアクションへの応答や、分析用途での活用などについては、ML モデルを REST エンドポイントとして展開することが推奨されます。これによりアプリケーションは、継続的に利用できるエンドポイントに対してリクエストを送信し、即座に予測内容を受信できるようになります。

Databricks は、Mosaic AI Model Serving と呼ばれるターンキーサービスを使用して、このプロセスを合理化します。Mosaic AI Model Serving は、Databricks によってファインチューニングまたは事前展開された Llama 2、MosaicML MPT、BGE などのモデルや、Azure OpenAI、AWS Bedrock、AWS SageMaker、Anthropic など、他のモデルプロバイダーのモデルを展開、管理、クエリ、そして監視するための統合サービスです。

Databricks ML Connector

Qlik Cloud Analytics は、Databricks Mosaic AI に対する 3 つの異なる接続手段を提供します。



- Databricks – Lakehouse からデータを抽出するために使用

- Databricks（Direct Access ゲートウェイ経由） – ゲートウェイを使用して Lakehouse からデータを抽出し、より高いセキュリティレベル（ファイアウォール、WAF など）の実装を行うために使用
- Databricks MLflow – この章で取り上げる範囲内で、Mosaic models と対話するために使用

  “MLFlow Analytic 接続 “ としてマークされた接続は、外部からアクセス可能な機械学習エンドポイントにデータを送信することができます。これらのエンドポイントはデータの計算を行い、その結果が返され Qlik Sense アプリケーションに組み込まれます。

  分析接続機能を Qlik Sense アプリケーションに統合する場合、データロードスクリプトとチャート式という 2 つの主要な方法があります。

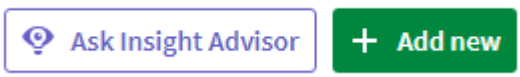
- データロードスクリプトは、アプリケーションがリロードされる際にデータを送信し、通常はより大きなデータセット（例えば、売上予測、顧客リストの解約データなど）を扱います。
- チャート式は、ユーザーの選択に基づいてデータをリアルタイムで送信し、その選択に基づいた結果を表示します。

  このプロセスにより、Qlik Analytics は外部の機械学習モデルを活用し、分析機能を強化することで、より堅牢で包括的なインサイトをユーザーに提供できるようになります。

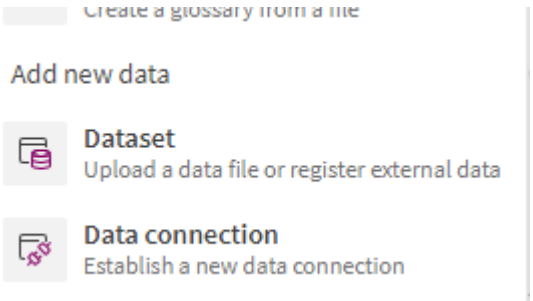
MLFlow 接続の作成

更新内容については、help.qlik.com から提供される最新のドキュメントをご覧ください。Databricks Mosaic ML 接続を作成する際は、次の手順に従ってください。

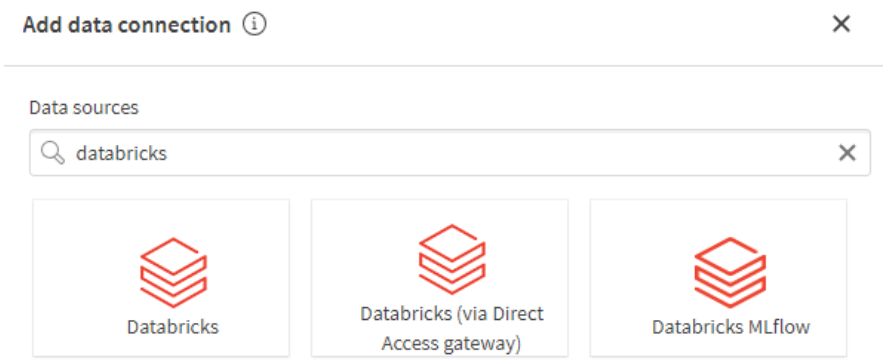
1. Qlik Cloud Analytics ハブへ移動し、[**+ Add new**] ボタンをクリックします



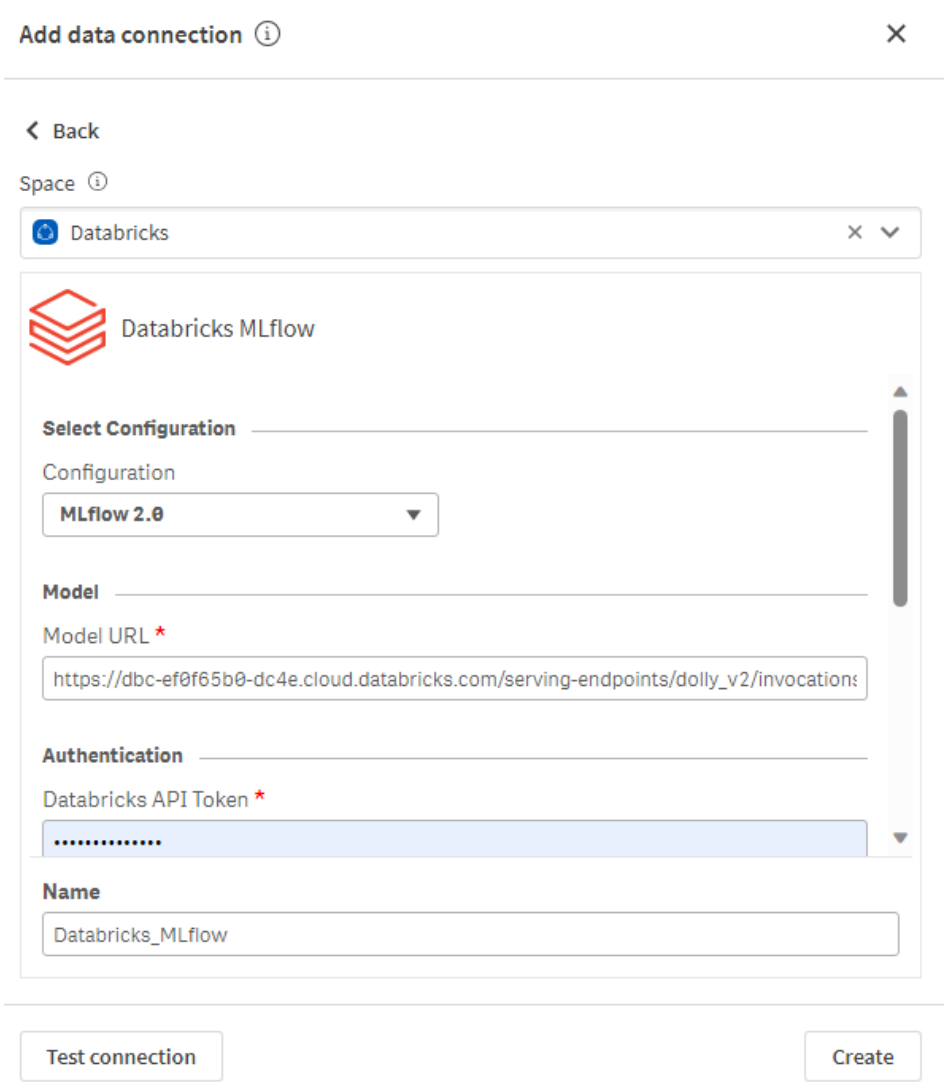
2. [**Data Connection**] オプションをクリックします



3. [**Data Sources**] ボックスで Databricks を検索し、オプション **Databricks MLflow** を選択します



4. [Model URL] フィールドに serving point URL を入力します（この情報は Databricks コンソールで入手できます）



5. Databricks API トークンに対して Databricks トークンを入力します（取得方法については、「Databricks 資格情報と接続の詳細の取得（Obtaining Databricks credentials and connection details）」をご覧ください）

6. 接続をテストし保存します

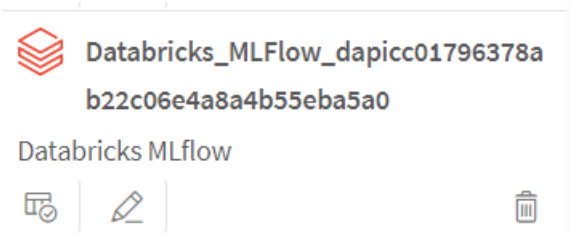
スクリプトレベルでの MLFlow 接続の使用

Databricks ML Connector セクションで説明したように、リロード時に MLFlow 接続を操作して、大量のデータを Databricks serving point へ送信し、その結果を Qlik Sense アプリのデータモデル内のフィールドとして使用することができます。データをエンドポイントに送信する場合には、事前にデータを Qlik Sense アプリへロードする必要があります。

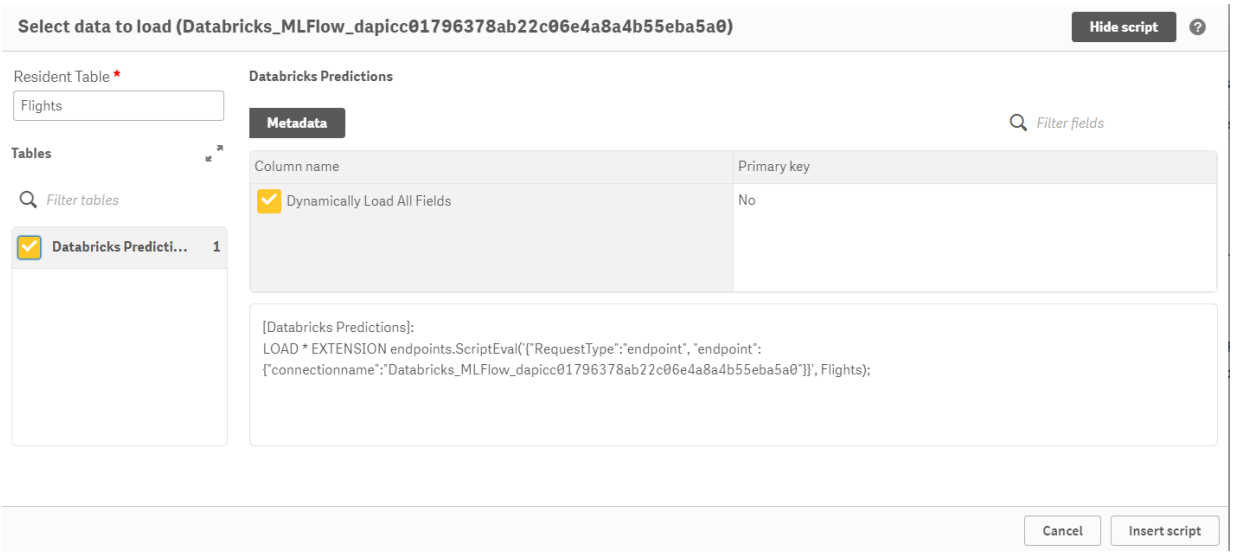
1. Qlik Sense アプリケーションにデータをインポートします。この例では、フライトキャンセルの予測モデルを利用して、フライトがキャンセルされるかどうかを予測しています。

```
1  Flights:
2  LOAD
3      *
4  FROM [lib://DataFiles/Flights.qvd](qvd);
5
```

2. ML_Flow ウィザードを使用してスクリプトを作成します。



3. **Resident Table** にテーブル名（step1 から）を入力し、**[Insert Script]** をクリックします。



4. アプリケーションをリロードし、Databricks Model の結果を含む新しいテーブルがデータモデルに追加されたかどうかを確認します。

5. アプリケーション内では、予測を通常のフィールドとして使用できます。

チャートレベルでの MLFlow 接続の使用

Databricks Mosaic ML Connector セクションで事前に説明したように、セグメント化されたデータ（ユーザーの選択に基づいてフィルターされたデータ）を Databricks serving point へ送信し、Qlik Sense オブジェクト（サードパーティの拡張機能を含む）内で結果を表示することで、MLFlow 接続をリアルタイムで利用することが可能になります。この接続を確立するための特定のシンタックスが存在します。Qlik Sense では独自の機能を提供しており、どの機能を選択するかは、接続先に送信されるデータの種類、および Qlik Engine がデータを送信前に集計するか、ディメンション値ごとに呼び出しを行うかによって決定されます。これらの関数に関するより包括的な説明については、Server Side Extensions Syntax ドキュメントでご確認頂けます。

一例として、以下では同じエンドポイントに接続して、モデルが予測を実行するために必要なパラメーターを渡し、1 件のキャンセル予測情報を取得しています。

追加リソース

このドキュメントで取り上げた具体的な項目の詳細については、以下のリストをご覧ください。

- Databricks データのローディング - https://help.qlik.com/en-US/cloud-services/Subsystems/ODBC_Connector_help/Content/Connectors_ODBC/Databricks/Databricks-connector.htm
- Databricks 接続の作成 - https://help.qlik.com/en-US/cloud-services/Subsystems/ODBC_Connector_help/Content/Connectors_ODBC/Databricks/Create-Databricks-connection.htm
- Databricks MLflow 接続の作成 - https://help.qlik.com/en-US/cloud-services/Subsystems/Hub/Content/Sense_Hub/LoadData/ac-databricks-mlflow-create.htm
- Databricks MLflow 分析ソース - https://help.qlik.com/en-US/cloud-services/Subsystems/Hub/Content/Sense_Hub/LoadData/ac-databricks-mlflow-overview.htm
- SQL Warehouses とは? - https://docs.databricks.com/en/compute/sql-warehouse.html
- Mosaic AI - https://www.databricks.com/product/machine-learning