ORACLE

Oracle Machine Learning for R

エンタープライズ・データに対してRを使用可能にする、

Oracle Machine Learningのコンポーネント

2023年4月、バージョン1.0

公開

本書の目的

本書では、Oracle DatabaseおよびOracle Autonomous Database用のOracle Machine Learning for Rの概要を説明しています。本書は、Oracle Machine Learning for Rのビジネス上の利点の評価と、データ・サイエンス・プロジェクトおよびITプロジェクトの計画立案を支援することのみを目的としています。

対象読者

本書は、Oracle DatabaseまたはOracle Autonomous Databaseに保存されたデータや、そこからアクセスできるデータを使用するデータの準備、機械学習、ソリューション・デプロイメントに対し、Rを大規模で使用することに関心のあるすべての方を対象としています。

目次

はじめに

Oracle Machine Learning for R（OML4R）は、Oracle DatabaseとOracle Autonomous Databaseで使用できるOracle Machine Learningのコンポーネントで、オープンソースのスクリプト言語および環境であるRをエンタープライズ・データに対して使用できるようにするものです。Oracle Machine Learning for Rは、Rを使用したエンド・ツー・エンドの分析プロセスにデータベース中心の環境を提供します。ユーザーは、データ探索とデータ準備のために、データベース内に存在するデータ上にあるオーバーロードされたR関数を呼び出します。ここでは、データベースを高パフォーマンスのコンピューティング環境として利用します。
また、Rユーザーが強力なインデータベース機械学習アルゴリズムをRから呼び出して、アプリケーションをサポートするユーザー定義のR関数をデプロイすることもできます。

企業向けのR

Rを通じてデータベースを高パフォーマンスのコンピューティング環境として利用するには、背後で透過的にSQLに変換されるオーバーロードされた機能とともに、R *data.frame*プロキシ・オブジェクトを通じてデータベース表やデータベース・ビューを操作します。これにより、データをデータベース内に保持したままで、使い慣れたR構文を使用してデータベース・データを操作できるため、より迅速かつ大規模にデータの探索、準備、分析を行うことが可能になります。

また、OML4Rでは自然なRインタフェースが提供されるため、インデータベース機械学習モデルを構築でき、それらのモデルをデータのスコアリングで使用できます。機械学習技術には、分類、リグレッション、クラスタリング、属性評価、異常検出、相関ルール、時系列、特徴抽出などが含まれます。インデータベース・アルゴリズムは、アルゴリズム別の自動データ準備、パーティション化されたモデル一式、および統合テキスト・マイニングをサポートします。Oracle Machine Learning for Rを使用すると、より多くのデータに対してさらに多くのモデルを構築し、大量データのスコアリングを素早く実行できます。データ・サイエンティストが生産性向上の恩恵を受けると同時に、エキスパートでないユーザーも強力なインデータベース・アルゴリズムを使用しやすくなります。

Oracle Autonomous Databaseでは、SQLやPythonと並んで、Oracle Machine Learning Notebooks経由でRを使用し、指定されたタスクに最適な言語を選択できます。OML4Rクライアント・パッケージを使用すると、RエンジンからオンプレミスのOracle DatabaseやOracle Database Cloud Serviceデータベースに接続できます。サードパーティ・クライアントからOracle Autonomous DatabaseへのOML4Rアクセスは、今後のリリースで導入されます。

埋込みR実行を使用することで、Rユーザーはユーザー定義のR関数を開発でき（サードパーティ・パッケージを含む）、それらを本番デプロイメント用のデータベース・スクリプト・リポジトリで管理できます。その後、これらのユーザー定義関数は、Oracle Autonomous Database上のSQLインタフェースおよびRESTエンドポイント経由、およびOracle Database上のSQLインタフェース経由で呼び出すことができます。たとえば、ユーザー定義のR関数をデータパラレルおよびタスクパラレルの方法で実行して、データベースによって起動および管理される複数のRエンジンを利用することで、ネイティブRモデルの大規模なスコアリングを実行することもできます。これらのユーザー定義関数の結果には、構造化された結果とイメージ結果の両方が含まれる可能性があります。

Rオブジェクトは、フラット・ファイル内に管理する代わりに、データ・ストア機能を使用してデータベース内に保存することもできます。データ・ストアは埋込みR実行と連携して、ユーザー定義関数に1つまたは複数のRオブジェクト（ネイティブR機械学習モデルなどの複雑なオブジェクトを含む）を渡します。

スクリプト・リポジトリ、データ・ストア、および埋込み実行機能により、データ・サイエンス・チーム内の共同作業が容易になるため、データ・サイエンティストからアプリケーション開発者へデータ・サイエンス作業成果を手軽に引き継ぐことができ、即時のデプロイメントが可能になります。

透過レイヤー

OML4Rの透過レイヤーにより、インデータベース・パフォーマンスを利用するための基盤が提供されます。プロキシ・オブジェクトとオーバーロードされたR関数（透過的にSQLに変換される）を使用して、データベース表の列索引、問合せ最適化、並列処理、インメモリ・キャッシング、表レベルのパーティション化を利用できるため、データベース表を操作する際のパフォーマンスを大幅に向上させることが可能です。

ここで、*ore.create*関数を使用してR *data.frame*（サンプルではdf）から表を作成し、使用したい表の名前（この場合は*'BOSTON'*）を指定する方法を見てみましょう。

```
ore.create(df, table = 'BOSTON')
ore.sync(table = 'BOSTON')

dim(BOSTON)
head(BOSTON)
summary(BOSTON)
min(BOSTON$age)
split(BOSTON, BOSTON$chas)
```

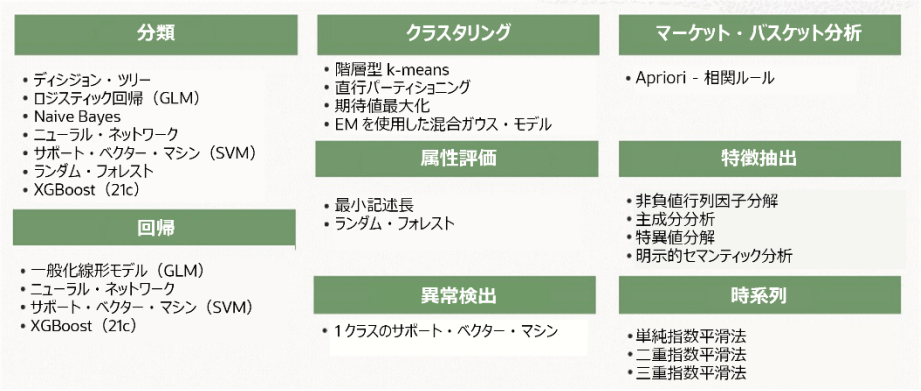
すでに表がデータベース内に存在する場合は、*ore.sync*関数を使用して、プロキシ・オブジェクトを取得したい対象の表またはビューの名前（もしくはSQL問合せ）を指定できます。このプロキシ・オブジェクト（サンプルではBOSTONと命名）はその後、*dplyr*パッケージからの関数を含む、*dim*、

head、*summary*、*split*などの使い慣れたR関数を呼び出すために使えるようになります。これらのオーバーロードされた関数により、インデータベース処理用に機能がSQLに変換されます。

インデータベースMLアルゴリズム

OML4Rは、Rの計算式仕様を使用して、自然なR APIをインデータベース・アルゴリズムに提供します。OML4Rは、以下のように、分類、リグレッション、クラスタリング、属性評価、異常検出、相関ルール、時系列、および特徴抽出のためのインデータベース・アルゴリズムをサポートします。インデータベース・アルゴリズムは、アルゴリズム別の自動データ準備、パーティション化されたモデル一式、および統合テキスト・マイニングをサポートします。

R APIを介して生成されたインデータベース・モデルは、SQL APIを使用したアクセス、個別管理のためにフラット・ファイルとしてエクスポート、他のOracle DatabaseインスタンスやAutonomous Databaseインスタンスへのインポート、およびOracle Machine Learning Servicesへのデプロイが可能です。



埋込みR実行

埋込みRの実行により、データベース環境によって起動および管理されるRエンジンで、ユーザー定義のR関数を実行できます。この機能により、ユーザー定義のR関数内にデータベース・データを自動的にロードできます。関数の結果には、イメージと構造化コンテンツの両方が含まれる可能性があり、構造化コンテンツは*data.frame*プロキシ・オブジェクト経由でアクセス可能な表として返されます。ユーザー定義のR関数は、データベース・スクリプト・リポジトリ内に保管して管理できます。

Oracle Autonomous Databaseでは、サード・パーティ・パッケージをユーザー定義関数内で使用して、モデルの構築、データのスコアリング、視覚化の生成などを実行できます。管理者は、conda環境を作成してOracle Object Storageにアップロードします。そこから、非管理者ユーザーは、Oracle Machine Learning Notebooksで直接使用したり、埋込み実行のSQLまたはREST呼出しで使用したりするために、特定のconda環境をダウンロードしてアクティブ化できます。Oracle Databaseでは、適切なデータベース・マシン・アクセス権限を持つ

ユーザーは、サードパーティ・パッケージをカスタム・インストールでき、ローカルのR環境から、または埋込み実行のSQL呼出しでそれらを使用できます。

また、埋込みRの実行インフラストラクチャでは、データ・パラレルおよびタスク・パラレルで関数を実行できるため、ネイティブのR機械学習モデルを使用したデータ・スコアリングなどの“Embarrassingly Parallel（自明な並列性）”のユースケースにも対応できます。

埋込みR実行により、ユーザー定義のR関数の本番デプロイが容易になるため、SQLインタフェース（Oracle Autonomous Databaseの場合はRESTインタフェース）から関数を呼び出してアプリケーション統合を実現できます。

- **doEval**はもっとも単純で、関数を単一のRエンジンで実行して、data.frameや、Rグラフィック・エンジンから生成されるイメージを含む関数の結果を返します。
- **tableApply**も単一のRエンジンで関数を実行しますが、プロキシ・オブジェクトによってR *data.frame*として参照されるデータを関数の最初の引数として渡します。
- **rowRpply**は、データを行チャンクにパーティション化することで並列処理を可能にします。データベース環境によって起動および制御される1つまたは複数のRエンジンを使用し、各チャンクに対してユーザー定義のR関数を実行します。
- **groupApply**もデータの並列処理を可能にします。データベース・データは、indexパラメータにより指定された列でパーティション化されます。parallel引数の指定に従って1つまたは複数のRエンジンを使用し、各パーティションに対してユーザー定義のR関数を実行します。
- **indexApply**は、タスクの並列処理を可能にするために、ユーザー定義のR関数を指定された回数実行し、indexの値を最初の引数として関数に渡し、parallel引数の指定に従って1つまたは複数のRエンジンを使用します。

以下のサンプルでは、R APIを使用してOracle Databaseでユーザー定義のR関数を呼び出しています。この関数は、*iris*データセットを使用してSpeciesを予測するランダム・フォレスト分類モデルを構築し、*“RF-model-iris”*というデータ・ストアにモデルを保管します。Oracle Databaseでは、*randomForest*パッケージがおのおののデータベース・サーバー・マシン・ノードのRエンジンにインストールされます。

```
buildRFmodel    <-    function(dat,dsname)    {
   library(randomForest)
   dat$Species <- as.factor(dat$Species)
   mod <- randomForest(Species ~ ., data=dat)
   ore.save(mod,name=dsname,overwrite = TRUE)
   TRUE
}
ore.connect(…database credentials…)
ore.scriptCreate('buildRFmodel', buildRFmodel)

ore.sync(table='IRIS') # ore.frameプロキシ・オブジェクトを取得
```

```
ore.tableApply(IRIS, FUN.NAME='buildRFmodel',
                     dsname= 'RF-model-iris',
                     ore.connect=TRUE)
```

関数を定義したら、*ore.connect*を使用してデータベースに接続し、ユーザー定義のR関数をデータベース・スクリプト・リポジトリに保存します。

*Autonomous Database*では、データベースは自動で接続されるため、明示的な接続手順がないことに注意してください。

次に、データベースにある*IRIS*表へのプロキシ・オブジェクトを取得します。これには、R変数*IRIS*を通じてアクセスできます。続いて、*tableApply*関数を使用します。この関数は、プロキシ・オブジェクト、スクリプト・リポジトリに保管された関数の名前、モデルを保管するデータ・ストア・インスタンスに使用するための名前、そしてデータ・ストア機能を使用しているために関数内でのデータベースへの接続を自動的に確立する状態を取ります。

以下のサンプルでは、REST APIとOracle Autonomous Database上の対応するSQL APIを使用して、ユーザー定義関数*buildRFmodel*を呼び出しています。

```
$ curl -i -X POST --header "Authorization:Bearer ${token}"
        --header 'Content-Type: application/json'
        --header 'Accept: application/json' \
        -d'{"input":"IRIS", "parameters":"{\"dsname\":\"RF-model- iris\"}"}' \
"${omlserver}/oml/api/r-scripts/v1/table-apply/buildRFmodel"

SELECT * FROM table(rqTableEval(
      inp_nam => 'IRIS',
      par_lst => '{"dsname":"RF-model-iris"}',
      out_fmt => 'XML',
      scr_name => 'buildRFmodel'));
```

SQLインタフェースを使用して、スクリプト・リポジトリにユーザー定義のR関数を保管することもできます。これにより、データ・サイエンティストとR開発者から本番環境の管理者への引継ぎが手軽に行えます。たとえば、以下のサンプルでは、*sys.rqScriptCreate*関数を使用して対応するスコアリング関数を保管しています。

```
BEGIN
  sys.rqScriptCreate('scoreDataRF',
      'function(dat, dsname) {
           library(randomForest)
           ore.load(name=dsname)
           dat$Prediction <- predict(mod, newdata = dat)
           dat[,c("Species","Prediction")]
       }',FALSE, TRUE); -- not global and enable overwrite
END;
```

スコアリングは本質的に自明な並列性のユースケースであるため、*ore.rowApply*を使用してスコアリング関数を呼び出して、一度に処理する行数（サンプルでは10）と使用するRエンジンの数（3）を指定できます。また、*FUN.VALUE*パラメータに、返す結果の構造体を指定します。このサンプルでは、結果として単一のプロキシ・オブジェクト*data.frame*を返します。

```
PRED = ore.rowApply(IRIS,
                    FUN.NAME = 'scoreDataRF',
                    rows = 10,
                    parallel = 3,
                    dsname = 'RF-model-iris',
                    ore.connect = TRUE,
                    FUN.VALUE = data.frame(Species=character(),
                                           Prediction=character(),
                                           stringsAsFactors=FALSE))
class(PRED)          # ore.frameプロキシ・オブジェクトを返す
ore.create(PRED,table = 'BATCH_SCORES') # persist as table
with(BATCH_SCORES, table(Species, Prediction))
```

これは、SQLおよびREST APIを使用して呼び出すこともできます。

データベース処理

Oracle Machine Learning for Rは、データベース内に存在するデータ上でのRスクリプトの実行において、データ・サイエンティストやより一般的なRユーザーをサポートします。これにより、使い慣れたR関数の使用をサポートしながら、クライアントへのデータ移動を回避します。OML4Rは、選択されたR関数をオーバーロードし、これらの関数を透過的にSQLへ変換します。これにより、データ移動がなくなると同時に、セキュリティが維持され、データベース表の列索引、問合せ最適化、並列処理、それにストレージレベルのパーティション化も利用できます。Oracle Machine Learning for Rは、並列処理、スケーラビリティ、およびメモリ最適化のために再設計

された広範囲にわたるインデータベース機械学習の技術とアルゴリズムをサポートし、マルチノード・クラスタや、Oracle Exadataなどの最適化されたハードウェアを利用します。

データ・サイエンティストとRユーザーが、インタラクティブにデータを分析してRスクリプトを開発する間、データと結果をデータベース内に保持できるので、データ移動による待機時間が短縮または解消され、さらにスケーラブルなデータ処理が実現します。ファイル抽出からデータをロードすることに慣れている場合は、OML4Rによって、データベース中心のアーキテクチャへの移行が容易になります。このようなアーキテクチャにより、コストと複雑さが軽減される一方で、生産性と信頼性が向上します。さらに、データをデスクトップからデータベースへ簡単に移動できます。

OML4Rは、選択されたオペレーティング・システムのOracle Autonomous Databaseバージョン19cおよび21c、およびOracle Databaseバージョン18c以降でサポートされます。OML4Rは、Oracle Autonomous Databaseサブスクリプション、Oracle Databaseライセンス、およびOracle Database Cloud Serviceと同梱されています。

結論

Oracle Autonomous DatabaseまたはOracle Databaseによって管理されるデータ、もしくはそれらにアクセスできるデータがある場合は、データベースのパフォーマンスとスケーラビリティを利用して、データ・サイエンティストや他のRユーザーによるデータの探索、準備、分析が可能になります。さらには、インデータベース・アルゴリズムでネイティブR APIを使用して、大規模に機械学習モデルを構築することもできます。ユーザー定義のR関数をデプロイしてシステムが提供する機能を利用し、Rエンジンを起動および制御することで、データをロードし、データのパラレル処理とタスクのパラレル処理を可能にします。これらはすべて、サードパーティ・パッケージを使用して実行できます。

Oracle Cloud Free TierでAlways Freeクラウド・サービスを使用して、Oracle Autonomous Database上でOracle Machine Learning for Rを無料でお試しください。
今すぐこちらでアカウントを作成してください。
また、Oracle Databaseで使用するためのOML4Rをこちらからダウンロードできます。

参考資料

*OML4R*のドキュメント

https://docs.oracle.com/en/database/oracle/machine-learning/oml4r

Oracle Machine Learning

データ・サイエンス・プロジェクトをサポートする製品ファミリで、機械学習、統計分析、データ準備、視覚化を提供する、*SQL/Python/R*用の*API*と次のコンポーネントで構成されています。*Oracle Data Miner*は、分析ワークフローを作成するためのドラッグ・アンド・ドロップ式ユーザー・インタフェースです。*Oracle Machine Learning Services*は、*REST*ベースのモデル管理とデプロイ、データとモデルの監視を提供します。*Oracle Machine Learning AutoML*ユーザー・インタフェースでは、ノーコードで機械学習モデルを作成できます。

https://oracle.com/data-science/machine-learning

ROracle

*ROracle*は、*DBI*に基づく*Oracle Database*接続用の*R*パッケージです。*Oracle Machine Learning for R*では、データベース接続時と埋込み*R*実行関数内で*ROracle*を使用します。*ROracle*は、データベース表での挿入、更新、および削除処理にも使用できます。

https://cran.r-project.org/web/packages/ROracle/index.html

https://www.oracle.com/database/technologies/roracle-downloads.html

Connect with us

+1.800.ORACLE1までご連絡いただくか、**oracle.com**をご覧ください。北米以外の地域では、**oracle.com/contact**で最寄りの営業所をご確認いただけます。

blogs.oracle.com facebook.com/oracle twitter.com/oracle