

ORACLE

SQLで始める機械学習入門

2020年3月

日本オラクル株式会社

Autonomous クラウド事業統括
クラウド・ソリューション本部

Safe harbor statement

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント（確約）するものではないため、購買決定を行う際の判断材料になさらないで下さい。

オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

文中の社名、商品名等は各社の商標または登録商標である場合があります。

Agenda

Oracle Machine Learning 環境へのアクセス

ハンズオン①：ある液体の品質を予測する

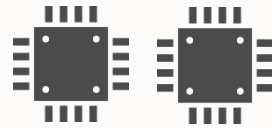
ハンズオン②：アソシエーション分析

まとめ：Autonomous Data Warehouseではじめるデータ活用基盤

機械学習による予測

INPUT

大量のデータ



情報処理

データの塊から
何かを予測する
予測モデルを作成

OUTPUT(機能=できること)

新しいデータに対し
予測モデルを適用して
活用する

データの準備

予測モデルを
作成・評価

実装
(サービスへの
組み込み)

Oracle Advanced Analyticsで利用可能なアルゴリズムと機能

分類 (Classification)



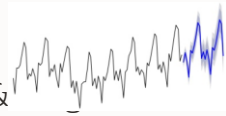
- 決定木 (Decision Tree)
- 単純ベイズ (Naïve Bayes)
- 一般線形モデル (GLM)
- ランダムフォレスト (Random Forest)
- サポートベクターマシン (SVM)
- 明示的セマンティック分析 (ESA)

クラスタリング (Clustering)



- 期待値最大化 (EM)
- k平均法 (k-means)
- 直行パーティショニング・クラスタリング (O-Cluster)

時系列 (Time Series)



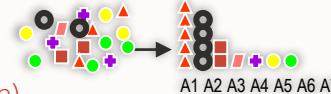
- Holt-Winters法、Regular & w/o trends & seasonal
- 単純指数平滑法、二重指数平滑法

回帰 (Regression)



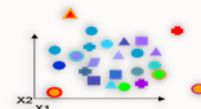
- 線形モデル(LM)
- 一般線形モデル(GLM)
- サポートベクターマシン (SVM)
- ステップワイズ線形回帰
- ニューラルネットワーク (Neural NW)

異常検出 (Abnormally Detection)



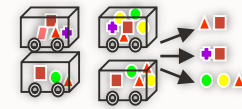
- 1クラス SVM

相関 (Association Rules)



- Apriori/ market basket

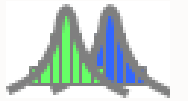
属性評価 (Attribute Importance)



- 最小記述長 (Minimum Description Length)
- CUR行列分解
- 教師なしペアワイズKLダイバージェンス

特徴抽出 (Feature Extraction)

- 明示的セマンティック分析 (ESA)
- Non-Negative Matrix Factorization (NMF)
- 特異値分解 Singular Value Decomposition (SVD)
- 予測成分分析 (PCA)
- Unsupervised Pair-wise KL Div



予測問合せ (Predictive Queries)

- クラスタリング問合せ
- 機能抽出問合せ
- 異常検出問合せ

統計関数 (Statistical Functions)

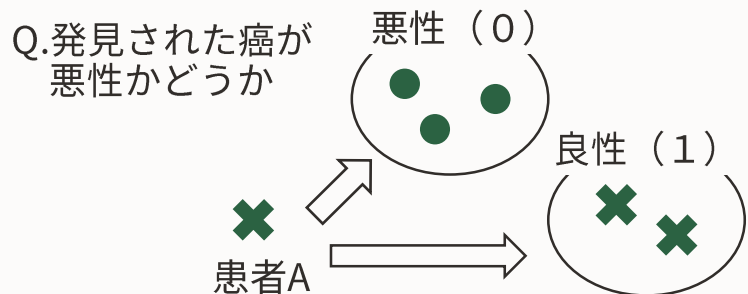
- 基本関数：最小、最大、メジアン、標準偏差、t検定、F検定、ピアソン分布、カイニ乗検定、分散分布(ANOVA)、等
- SQL関数：行列の選択、条件式、結合、等



教師あり学習と教師なし学習

分類

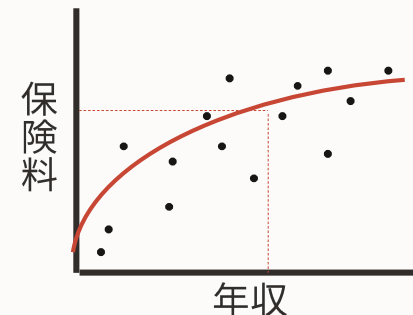
どこに属するかを予測する



回帰分析

数字（連続値）を予測する

Q.年収から保険料を予測する



凡例)

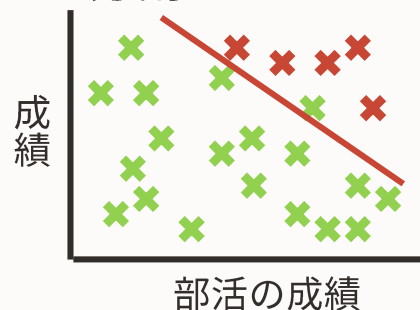
教師あり学習

教師なし学習

クラスタリング

グループに分割する

Q.優等生を見つける



異常検出

異常を検知する

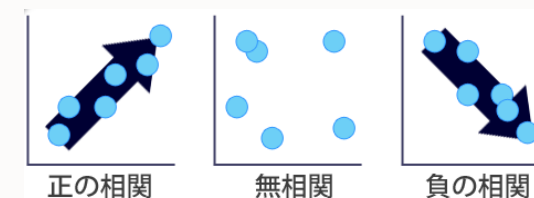
Q.カードの不正利用をみつける



相関分析

関係性が強いものを見つける

Q.年収に一番影響するものは何か



Oracle Machine Learning とは

- Oracle Autonomous Database (※1) に付属する、SQL Notebookのインタフェース (Apache Zeppelinベース)
- データ・サイエンティストがOracle Autonomous Databaseで機械学習を実行するのに役立ちます



※1 Oracle Autonomous Database は、Oracle Cloud 環境で利用できる、世界初の自律型データベース・クラウドです



Autonomous Databaseが実現する世界

自律型データベースに対するオラクルのビジョン

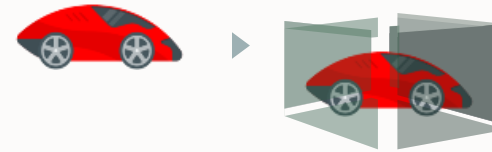
予め構成・最適化
・テスト済み



自動的に
モニタリング



自動的にバックアップ



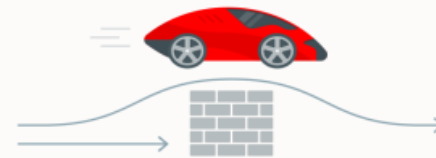
自動的に
オンラインでパッチ適
用



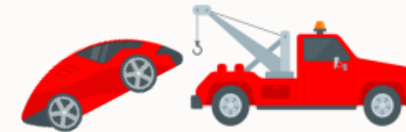
自動的にスケール



自動的に障害回避



自動的に
エラーハンドリング



自動的に
セキュアな構成



自動的に
パフォーマンス診断



自動的に最適化



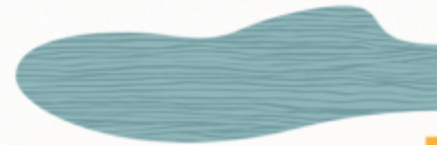
自動的に
テスト実行



自動的に
移行とデータロード



Oracle Machine Learning Notebook 環境へのアクセス



Oracle Machine Learning Notebook 環境 サインイン

ORACLE クラウド・インフラストラクチャ



SIGN IN

Database name:
MYADW

Oracle Machine Learningデータベースユーザー資格証明を使用してサインインします。

ユーザー名 *


パスワード *

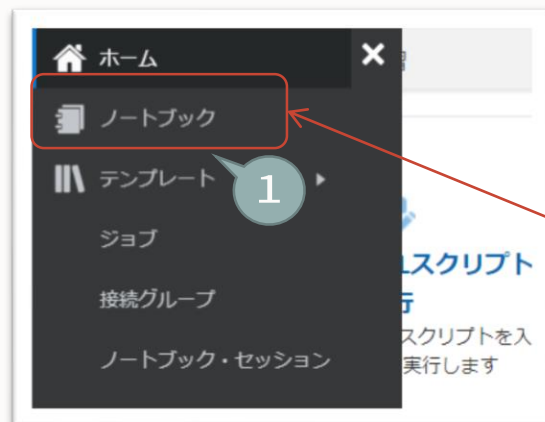
[サインイン](#)

ユーザー名・パスワードを入力し
サインインします

Oracle Machine Learning Notebook環境へ
サインインができるようになるまでの手順は、
Autonomous Data Warehouseの管理画面
から、数クリックで実施できます

新しいNotebookを作成

- 1 「ノートブック」をクリック
(ホーム画面、もしくは左上の  から「ノートブック」へのリンクが表示されます)



新しいNotebookの作成

2 「作成」をクリック



ノートブック

編集 + 作成 複製 テンプレートとして保存 削除 インポート バージョン

名前	コメント	最終更新日	更新者	接続グループ
Anomaly Detection	This notebook shows how to det...	2019/02/06 12:37	ORAML	Global

3 ノートブックの名前を入力し、OKをクリック



ノートブックの作成 ×

名前
液体の品質予測

コメント

接続
Global

OK 取消

新しいNotebookを作成：編集画面

インタープリタ・
バインディング(次頁参照)

接続ユーザ名
接続の状態

センサーデータの異常検出

パラグラフ (段落)

パラグラフ (段落)

+ Add Paragraph

このパラグラフに対する各種設定

- パラグラフのタイトル記述
- パラグラフを上/下に
- パラグラフの削除

など

パラグラフ間に
マウスを合わせると
「Add Paragraph」
が表示される
(クリックで
パラグラフ追加)

接続ユーザ名

接続の状態

Notebook全体を実行
※%sql, %scriptのように
異なるインタープリタを含む
場合はこのボタンは使用せず、
パラグラフを1つずつ実行する

パラグラフ内を実行
(Shift+Enterキーでも可)

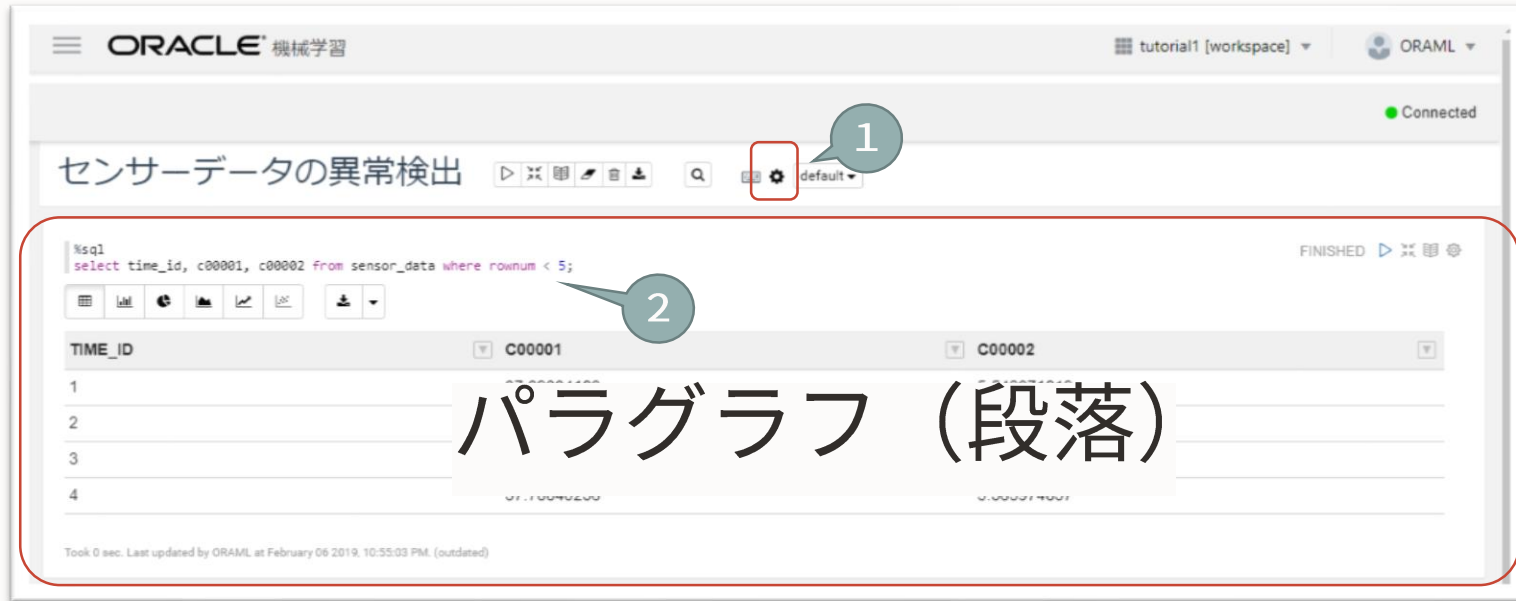
このパラグラフに
対する各種設定

- パラグラフのタイトル記述
- パラグラフを上/下に
- パラグラフの削除

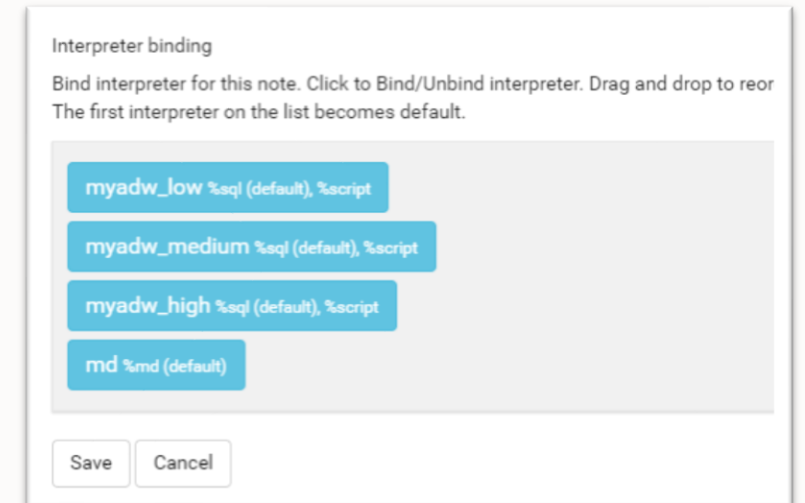
など



新しいNotebookを作成：編集画面



- 1 インタープリタ・バインディングの設定が確認・変更できる
(テンプレートからのコピー時や、JSONインポート時は、接続先が選択されていないので明示的に選択する必要がある)
※ブラウザの上部に表示されます



- 2 パラグラフ先頭で指定できるのは3種類
 - **%sql**: 単一のSQL用、結果表示は文字列だけでなくグラフ化が可能
 - **%script**: 複数のSQL用
 - **%md**: テキスト用、マークダウン方式でテキストを記載する

補足：既存のNotebookの複製・削除

- 1 Notebook一覧の頁で、削除したいNotebookをクリックし、選択状態にする（背景色が黄色になる）
- 2 「複製」または「削除」をクリック

名前	コメント	最終更新日	更新者	接続グループ
Anomaly Detection	This notebook shows how to det...	2019/02/06 12:37	ORAML	Global
Anomaly Detection_fromTemplate	This notebook shows how to det...	2019/02/06 8:01	ORAML	Global
Association Rules	Notebook to show the use of Ass...	2019/01/30 3:32	ORAML	Global
Association Rules1	Notebook to show the use of Ass...	2019/02/01 9:07	ORAML	Global

※ 「複製」 や 「テンプレートとして保存」 操作も同様の手順です

ハンズオン①：ある液体の品質を予測する



ある液体の品質を予測する



良品？



通常品？



使用するデータ

LIQUID表の内容 (レコード数: 4,898)

予測したい値 (目的変数)

- 品質(1=良品, 0=普通)を示す値
- 予測モデル作成時に**TARGET_ID**として指定する

ID	FIXED_AC IDITY	VOLATI LE_ACID ITY	CITRIC_AC ID	RESIDU AL_SUG AR	FREE_SUL FUR_DIOX IDE	TOTAL_S ULFUR_D IOXIDE	DENSITY	PH	SULPHATE S	ALCOHO L	QUALITY	
1	3.8	0.31	0.02	11.1	0.036	20	114	0.9925	3.75	0.44	12.4	1
2	3.9	0.225	0.4	4.2	0.03	29	118	0.989	3.57	0.36	12.8	1
3	4.2	0.17	0.36	1.8	0.029	93	161	0.99	3.65	0.89	12	1
4	4.2	0.215	0.23	5.1	0.041	64	157	0.9969	3.42	0.44	8	0

一意な値

- Primary Keyに相当(PK制約は必須ではない)
- 予測モデル作成時に**CASE_ID**として指定する

各属性 (説明変数) とその値

- pH値、アルコール度数、クエン酸濃度など
全部で11個の属性

各属性の値から、
品質 (QUALITY列の値) を
予測するモデルを作成する

※データは、下記より一部の値を加工したものを利用しています
P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.
Modeling wine preferences by data mining from physicochemical properties.
In Decision Support Systems, Elsevier, 47(4):547-553, 2009.







データを 確認する

件数などの基本的な情報

- 1 パラグラフ内の先頭行に`%sql`、次の行に実行したいSQLを入力し、 をクリックし実行します


```
%sql  
select count(*) from liquid;
```

FINISHED    

- 2 実行結果が表示されます

※  の代わりに、Shift + Enterキーでも実行できます

```
%sql  
select count(*) from liquid;
```

FINISHED    

COUNT(*)
4898

データを 確認する

件数などの基本的な情報

- 3 他のSQLも実行してみます
今回予測したいQUALITY列の値の種類・それぞれの件数を確認します

```
%sql
select quality, count(*) from liquid group by quality;
```

FINISHED ▶ ⌵ 📖 ⚙️

📊 📈 📉 📊 📈 📉 📊 📈 📉

QUALITY	COUNT(*)
0	1640
1	3258

データを 確認する

件数などの基本的な情報

- 1 新しいパラグラフが自動的に下に作成されており、そのパラグラフを使用します
先頭行に`%sql`、2行目に `select * from liquid;` を入力し、実行します

Took 0 sec. Last updated by ORAML at February 17 2019, 10:41:59 PM. (outdated)

```
%sql
```

READY ▶ ✖ 📄 ⚙

- 2 実行結果が表示されます

```
%sql  
select * from liquid;
```

FINISHED ▶ ✖ 📄 ⚙

📄 📊 📈 📉 📊 📉 📄 ⌵

ID	FIXED_ACIDITY	VOLATILE_ACIDITY	CITRIC_ACID	RESIDUAL_SUGAR	CHLORIDES	FREE_SULFUR_DIOXIDE
1	3.8	0.31	0.02	11.1	0.036	20
2	3.9	0.225	0.4	4.2	0.03	29

※件数が多い場合は`where rownum < 100`のように絞るか、`select * from liquid sample (10);`のように、サンプル指定します

データを確認する

SQL結果をグラフ表示に変更

- 1 表示方式を散布図（Scatter Chart）に変更します
前ページで実行したパラグラフ内で、**散布図**のアイコンをクリックし、次に **Settings** をクリックします

The screenshot shows a data visualization tool interface. At the top, there is a SQL query editor with the text `%sql select * from liquid;`. Below the editor is a toolbar with various chart icons. A callout bubble with the number '1' points to the scatter plot icon. To the right of the toolbar is a 'settings' button, also indicated by a callout bubble with the number '1'. Below the toolbar, there is a section labeled 'All fields:' containing a list of field names: ID, FIXED_ACIDITY, VOLATILE_ACIDITY, CITRIC_ACID, RESIDUAL_SUGAR, CHLORIDES, FREE_SULFUR_DIOXIDE, TOTAL_SULFUR_DIOXIDE, DENSITY, PH, SULPHATES, ALCOHOL, and QUALITY. Below this list, there are input fields for 'xAxis', 'yAxis', 'group', and 'size'. The 'xAxis' field contains 'PH' and is indicated by a callout bubble with the number '2'. The 'yAxis' field contains 'ALCOHOL' and is also indicated by a callout bubble with the number '2'.

- 2 xAxis(X軸) に **PH**, yAxis(Y軸) に **ALCOHOL** をドラッグします

データを 確認する

SQL結果をグラフ表示に変更

列同士の相関をグラフで可視化しています

X軸にpH列の値、Y軸にALCOHOL列の値がプロットされています
軸に指定する列は、列名のドラッグ操作で変更できます



データを確認する

最小値・最大値・平均・標準偏差・NULLの数を確認

min	最小値
max	最大値
avg	平均
stddev	標準偏差
count(1) - count(列名)	NULLの数

- 1 次は %script を使用してみます
1行目に**%script** 2行目以降にSQLを入力し、実行します
→ テキスト形式で出力されます

```
%script
select min(alcohol) as min, max(alcohol) as max, avg(alcohol) as avg,
       stddev(alcohol), count(1) - count(alcohol) as nullcount
from liquid;
```

MIN	MAX	AVG	STDDEV(ALCOHOL)	NULLCOUNT
8	14.2	10.51451612903225806451612903225806451613	1.23088512517933570080389161264114642998	0


```
%script
select min(ph), max(ph), avg(ph), stddev(ph), count(1) - count(ph) as nullcount from liquid;
```

MIN(PH)	MAX(PH)	AVG(PH)	STDDEV(PH)	NULLCOUNT
2.72	3.82	3.18828185723051748823890366128042544488	0.15113913312416952995000668678586927158	9



データを準備する

LIQUID表のデータをTRAIN用とTEST用に分割

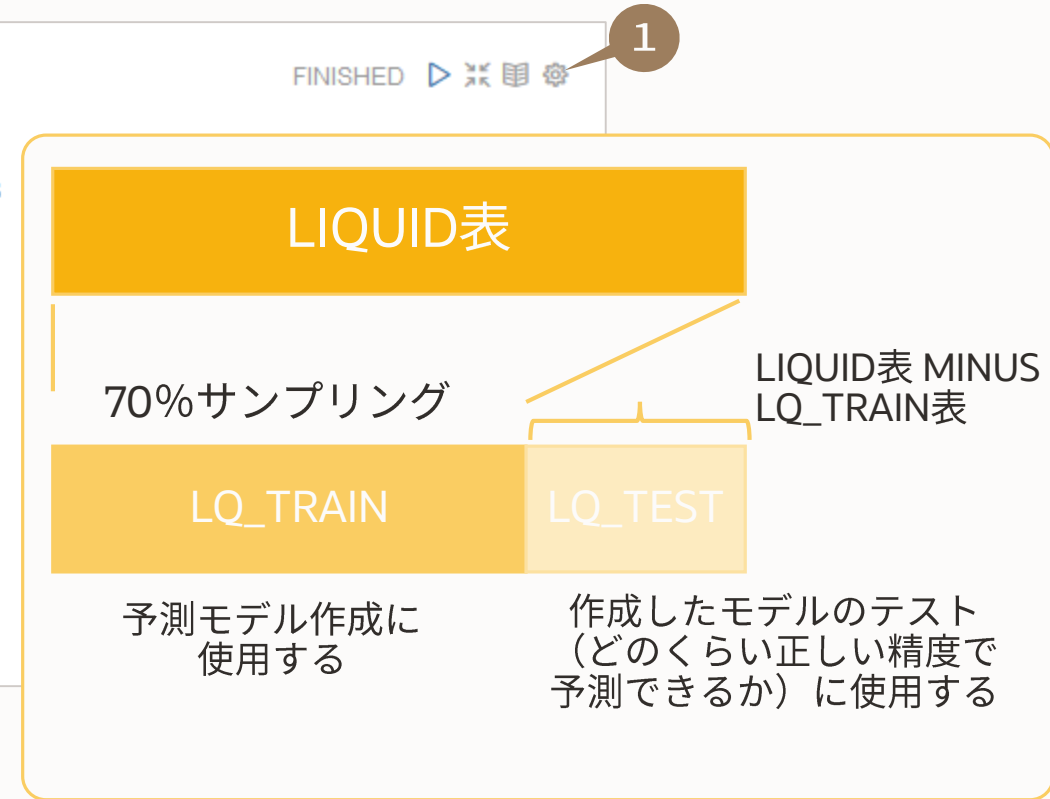
- 1 %script を使用し、以下のSQLを実行します
 パラグラフにタイトルを付けるには、 をクリックして **Show Title** を指定し、入力します

LIQUID表とLQ_TEST表に分割する ←—— パラグラフのタイトル

```
%script
create table lq_train as select * from liquid sample(70);
create table lq_test as select * from (select * from liquid minus select * from lq_train);
select count(*) from lq_train;
select count(*) from lq_test;
```

LIQUID表を SAMPLE指定を用いて
分割するSQLを実行

```
Table LQ_TRAIN created.
-----
Table LQ_TEST created.
-----
COUNT(*)
          3394
-----
COUNT(*)
          1504
-----
```



データを準備する

データをTRAIN用とTEST用に分割する理由

教師あり学習では、各データにラベル（予測したい値＝目的変数）が付与されています

分類の場合：「良品 or 不良品」「契約した or 契約しなかった」「晴れ or 雨 or 曇り」

全データのうち一部を使用してモデルを作成し、残りのデータでモデルの精度を評価します

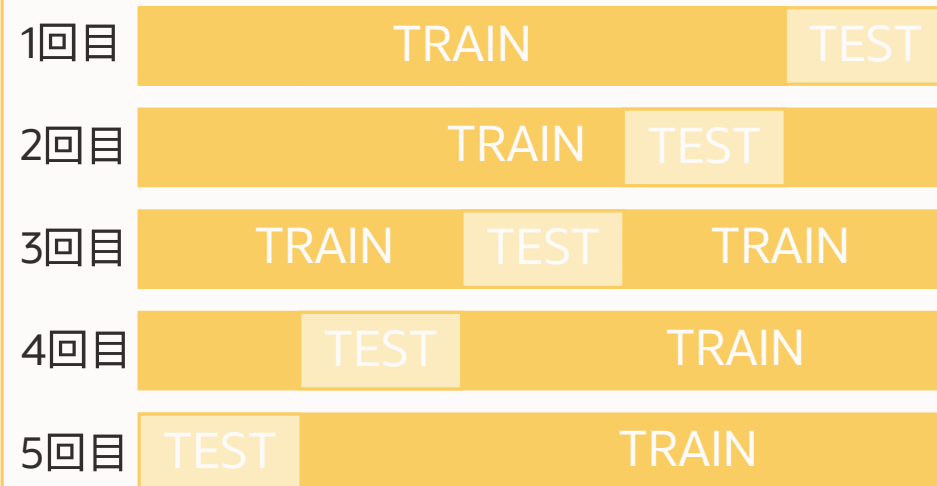
本ハンズオンではホールドアウト法で評価します

ホールドアウト法



※1：データをN個に分けて、1回目はそのうち1つをTEST用、残りのN-1個をTRAIN用として使用
2回目は別の1つをTEST用、残りのN-1個をTRAIN用として使用・・・をN回繰り返し、各回で測定した精度の平均を取る

クロスバリデーション (※1)



モデル生成・評価に使用するオブジェクト

本ハンズオンで使用するオブジェクトの名前

Oracle Advanced Analyticsを使用した機械学習では、使用するアルゴリズムのパラメータ設定や、結果の評価データ格納先として、テーブルを使用します

テーブル名やモデル名は任意ですが（実行時に指定する）、ハンズオンでは以下の名前で作成します
同じ名前で再作成する場合は、事前にこれらを削除（テーブルの削除と、モデルの削除）します

作成するモデルの名前： **LQ_MODEL_<アルゴリズム>**

	テーブルの名前
モデルの設定（パラメータ格納）	LQ_MODEL_<アルゴリズム>_SETTING
モデルの適用結果を格納（適用結果表）	LQ_MODEL_<アルゴリズム>_APPLY_RESULT
モデルの評価に使用（リフト）	LQ_MODEL_<アルゴリズム>_LIFT
モデルの評価に使用（混同行列）	LQ_MODEL_<アルゴリズム>_CONFUSION_MATRIX

※2回目以降に同じ名前で実行する場合

ランダムフォレスト：初期化

```
%script
drop table lq_model_rf_setting;
drop table lq_model_rf_apply_result purge;
drop table lq_model_rf_lift purge;
drop table lq_model_rf_confusion_matrix purge;
call dbms_data_mining.drop_model('LQ_MODEL_RF');
```

ランダム・フォレスト：モデル作成

モデル作成で使用するパラメータの設定

- 1つのパラグラフ内に、以下を入力し、実行します。パラグラフのタイトルを入力します。

パラグラフのタイトル入力 1

ランダムフォレスト：モデル作成で使用するパラメータの設定

FINISHED ▶ ❌ 📖 ⚙️

```
%script
create table lq_model_rf_setting (setting_name varchar2(30), setting_value varchar2(4000));
insert into lq_model_rf_setting values ('ALGO_NAME', 'ALGO_RANDOM_FOREST'); -- ランダムフォレストを指定
insert into lq_model_rf_setting values ('PREP_AUTO', 'ON'); -- 自動データ準備をONに
```

Table LQ_MODEL_RF_SETTING created.

1 row inserted.

1 row inserted.

パラメータ設定

ALGO_NAME に、**ALGO_RANDOM_FOREST** (ランダムフォレスト) を設定
PREP_AUTO に、**ON** を設定
設定可能な「パラメータ名」と「設定値」は、製品ドキュメント参照

※ ここから先の作業、シングルコーテーション ' で囲まれる箇所では、
全て大文字で入力することを推奨します (一部の設定項目は大文字小文字の違いを認識するため)

ランダム・フォレスト：モデル作成

QUALITY列の値を予測する分類モデルを作成する

- 2 次の新しいパラグラフ内に、以下を入力し、実行します。パラグラフのタイトルを入力します。

The screenshot shows a SQL script execution in a tool. The script is titled "ランダムフォレスト：モデル作成" and is in a state of "FINISHED". The script content is as follows:

```
%script
call dbms_data_mining.create_model('LQ_MODEL_RF', 'CLASSIFICATION', 'LQ_TRAIN', 'ID', 'QUALITY', 'LQ_MODEL_RF_SETTING');
Call completed.
```

Callouts explain the parameters in the script:

- 使用するデータのテーブル名 (Table name of the data to be used) points to 'LQ_TRAIN'.
- ケースID (一意キー) (Case ID (unique key)) points to 'ID'.
- 予測したい列 (Column to be predicted) points to 'QUALITY'.
- モデル名 (Model name) points to 'LQ_MODEL_RF'.
- マイニング機能の名前 (Mining function name) points to 'CLASSIFICATION'.
- パラメータ格納表 (Parameter storage table) points to 'LQ_MODEL_RF_SETTING'.

A dashed box contains the following text:

CLASSIFICATION：（分類）モデルを指定
指定可能な値は、製品ドキュメント参照

ランダム・フォレスト：作成したモデルを評価

TEST表のデータを使用して、作成したモデルを評価

3 次の新しいパラグラフ内に、以下を入力し、実行します。パラグラフのタイトルを入力します。

ランダムフォレスト：モデルを評価 FINISHED ▶

```

%script
-- TEST表のデータにモデルを適用
call dbms_data_mining.apply('LQ_MODEL_RF', 'LQ_TEST', 'ID', 'LQ_MODEL_RF_APPLY_RESULT');
-- LIFTを計算
call dbms_data_mining.compute_lift('LQ_MODEL_RF_APPLY_RESULT', 'LQ_TEST', 'ID', 'QUALITY', 'LQ_MODEL_RF_LIFT', '1');
-- 混同行列を計算
declare
  v_accuracy number;
begin
  dbms_data_mining.compute_confusion_matrix(v_accuracy, 'LQ_MODEL_RF_APPLY_RESULT',
    'LQ_TEST', 'ID', 'QUALITY', 'LQ_MODEL_RF_CONFUSION_MATRIX');
  dbms_output.put_line('*** model accuracy ***: ' || round(v_accuracy,4));
end;
/
-- 混同行列を表示
select * from lq_model_rf_confusion_matrix;
    
```

評価用データのテーブル名

ケースID (一意キー)

予測したい列

ポジティブクラスを指定 (後述)

モデル名

変数名(任意)

v_accuracy

number;

begin

dbms_data_mining.compute_confusion_matrix(v_accuracy, 'LQ_MODEL_RF_APPLY_RESULT',

'LQ_TEST', 'ID', 'QUALITY', 'LQ_MODEL_RF_CONFUSION_MATRIX');

dbms_output.put_line('*** model accuracy ***: ' || round(v_accuracy,4));

end;

/

-- 混同行列を表示

select * from lq_model_rf_confusion_matrix;

テーブル名：適用結果表

テーブル名：リフト

テーブル名：混同行列



ランダム・フォレスト：作成したモデルを評価

TEST表のデータを使用して、作成したモデルを評価

前ページ実行結果

```
Call completed.
-----
Call completed.
-----
**** model accuracy ****: .8012
PL/SQL procedure successfully completed.
```

v_accuracyの値（モデルの正解率）

ACTUAL_TARGET_VALUE	PREDICTED_TARGET_VALUE	VALUE
0	0	307
0	1	200
1	1	898
1	0	99

混同行列表の
SELECT結果

混同行列表のSELECT結果
(左と同じ内容をマトリックス形式で表示)

```
%script
select * from lq_model_rf_confusion_matrix
pivot ( max(value) for PREDICTED_TARGET_VALUE in (
    '1' as "予想=1",
    '0' as "予想=0"
)
)
order by ACTUAL_TARGET_VALUE desc
;
```

ACTUAL_TARGET_VALUE	予想=1	予想=0
1	898	99
0	200	307

SVM：モデル作成と評価

パラグラフをコピーして編集後、実行

- 4 先の手順①～③で作成した3つのパラグラフを、それぞれコピーして編集します。**Clone paragraph** をクリックします。編集が終わったら実行します。
(パラグラフをコピーした際に、間違いがないよう、パラグラフのタイトルも編集することをおすすめします)

ランダムフォレスト：モデル作成で使用するパラメータの設定

```
%script
create table lq_model_rf_setting (setting_name varchar2(30),setting_value varchar2(4000));
insert into lq_model_rf_setting values ('ALGO_NAME', 'ALGO_RANDOM_FOREST'); -- ランダムフォ
insert into lq_model_rf_setting values ('PREP_AUTO', 'ON'); --自動データ準備をONに
```

Table LQ_MODEL_RF_SETTING created.

1 row inserted.

1 row inserted.

Took 0 sec. Last updated by ORAML at February 18 2019, 10:32:15 PM.

ランダムフォレスト：モデル作成

```
%script
call dbms_data_mining.create_model('LQ_MODEL_RF', 'CLASSIFICATION', 'LQ_TRAIN', 'ID', 'QUALI
```

Call completed.

Took 3 sec. Last updated by ORAML at February 18 2019, 10:32:23 PM.

ランダムフォレスト：モデルを評価

```
%script
-- TEST表のデータにモデルを適用
```

「モデル作成で使用するパラメータの設定」
「モデル生成」
「モデルを評価」
を、全て、Clone Paragraphでコピーします。

場所はこの順になるよう適宜並び替えます
(Move Up / Move Downでパラグラフを移動)

編集箇所は、次の2ページを参照ください。

SVM：モデル作成と評価

編集箇所：赤枠内を、以下の内容に編集します

SVM：初期化（※2回目以降に同じ名前で行う場合。今回は不要）

FINISHED ▶ ❌ 📄 ⚙️

```
%script
drop table lq_model_svm_setting;
drop table lq_model_svm_apply_result purge;
drop table lq_model_svm_lift purge;
drop table lq_model_svm_confusion_matrix purge;
call dbms_data_mining.drop_model('LQ_MODEL_SVM');
```

変更のポイント

- 使用するオブジェクト名を RM → SVM に変更
- SVMアルゴリズムを使うことを指定

SVM：モデル作成で使用するパラメータの設定

FINISHED ▶ ❌ 📄 ⚙️

```
%script
create table lq_model_svm_setting (setting_name varchar2(30),setting_value varchar2(4000));
insert into lq_model_svm_setting values ('ALGO_NAME', 'ALGO_SUPPORT_VECTOR_MACHINES'); --SVM
insert into lq_model_svm_setting values ('PREP_AUTO', 'ON'); --自動データ準備をONに
```

SVM：モデル作成

FINISHED ▶ ❌ 📄 ⚙️

```
%script
call dbms_data_mining.create_model('LQ_MODEL_SVM', 'CLASSIFICATION', 'LQ_TRAIN', 'ID', 'QUALITY', 'LQ_MODEL_SVM_SETTING');
```

Call completed.

SVM：モデル作成と評価（実行するスクリプト）

SVM：モデルを評価

FINISHED ▶

```
%script
-- テスト表のデータにモデルを適用
call dbms_data_mining.apply('LQ_MODEL_SVM','LQ_TEST','ID','LQ_MODEL_SVM_APPLY_RESULT');
-- LIFTを計算
call dbms_data_mining.compute_lift('LQ_MODEL_SVM_APPLY_RESULT','LQ_TESTV','ID','QUALITY','LQ_MODEL_SVM_LIFT','1');
-- 混同行列を計算
declare
  v_accuracy    number;
begin
  dbms_data_mining.compute_confusion_matrix(v_accuracy,'LQ_MODEL_SVM_APPLY_RESULT','LQ_TEST',
  'ID','QUALITY','LQ_MODEL_SVM_CONFUSION_MATRIX');
  dbms_output.put_line('**** model accuracy ****: ' || round(v_accuracy,4));
end;
/
-- 混同行列を表示
select * from lq_model_svm_confusion_matrix;
```

使用するオブジェクト名をRM → SVM に
変更しています

Call completed.

Call completed.

**** model accuracy ****: .7533

PL/SQL procedure successfully completed.

ACTUAL_TARGET_VALUE	PREDICTED_TARGET_VALUE	VALUE
1	0	91
0	0	227
0	1	280
1	1	906

作成したモデルを評価する

正解率

どのくらい正しく予測できたかを、テストした件数全体で計算
v_accuracy として手順の中で計算・出力している

dbms_data_mining.compute_confusion_matrix のOUT変数
次ページの混同行列では、（グレー部分 ÷ テストした全ての件数）が該当

```
DBMS_DATA_MINING.COMPUTE_CONFUSION_MATRIX (  
    accuracy                               OUT NUMBER,  
    apply_result_table_name               IN  VARCHAR2,  
    target_table_name                     IN  VARCHAR2,  
    case_id_column_name                   IN  VARCHAR2,  
    target_column_name                     IN  VARCHAR2,  
    confusion_matrix_table_name           IN  VARCHAR2,  
    score_column_name                     IN  VARCHAR2 DEFAULT 'PREDICTION',  
    ...  
    score_criterion_type                   IN  VARCHAR2 DEFAULT 'PROBABILITY');
```

作成したモデルを評価する

混同行列 (Confusion Matrix)

分類器の性能を評価するのに、正しく識別できた件数・誤って識別した件数を比較するためにまとめる行列

実際の値を縦に並べ、横方向が予測されたクラス (の形式で表現されることが多い)

適合率 (PRE) と再現率 (REC) から計算されるF値を、性能指標としてよく利用される

$$F\text{値} = 2 \frac{PRE \cdot REC}{PRE + REC}$$

混同行列 (2クラスの場合)

	予測された クラス=1	予測された クラス=0
実際=1	真陽性 (TP)	偽陰性 (FN)
実際=0	偽陽性 (FP)	真陰性 (TN)

再現率 (REC) の
計算に使用

適合率 (PRE) の
計算に使用

```
%script
select * from lq_model_rf_confusion_matrix
pivot ( max(value) for PREDICTED_TARGET_VALUE in (
    '1' as "予想=1",
    '0' as "予想=0")
)
order by ACTUAL_TARGET_VALUE desc
;
```

混同行列表のSELECT結果
(pivotで行列形式で表示)

```
ACTUAL_TARGET_VALUE  予想=1  予想=0
1                    898    99
0                    200   307
-----
```

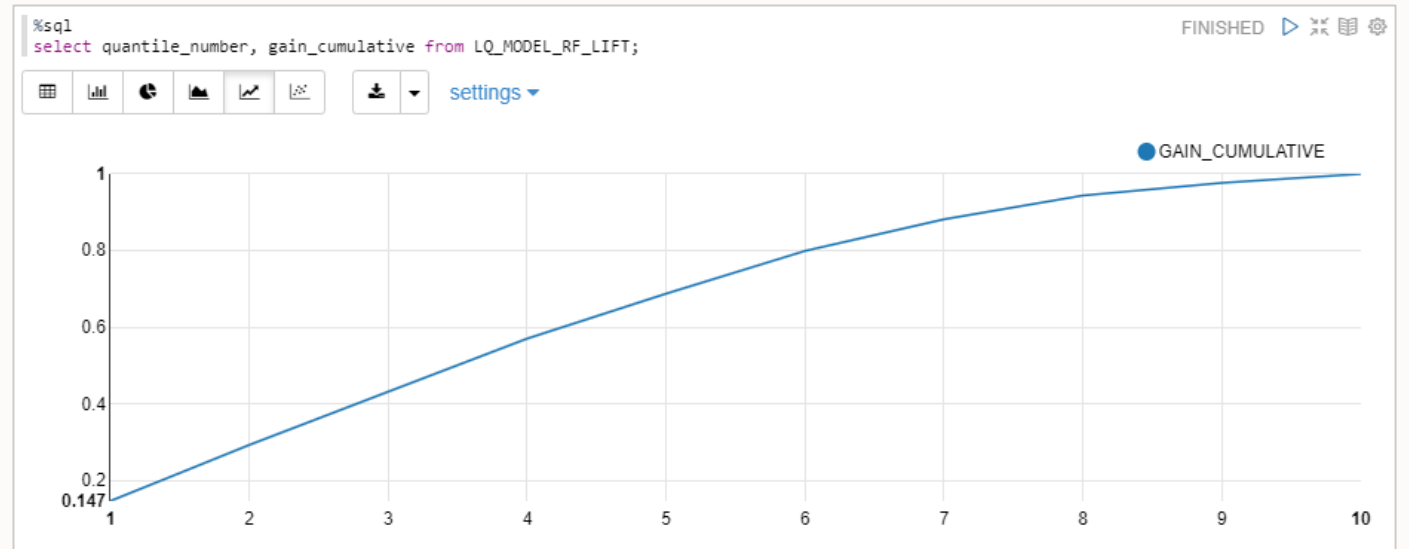
作成したモデルを評価する

リフト

ランダムに生成された予測と比較して、分類モデルの予測がどの程度信頼できるかを計算したもの

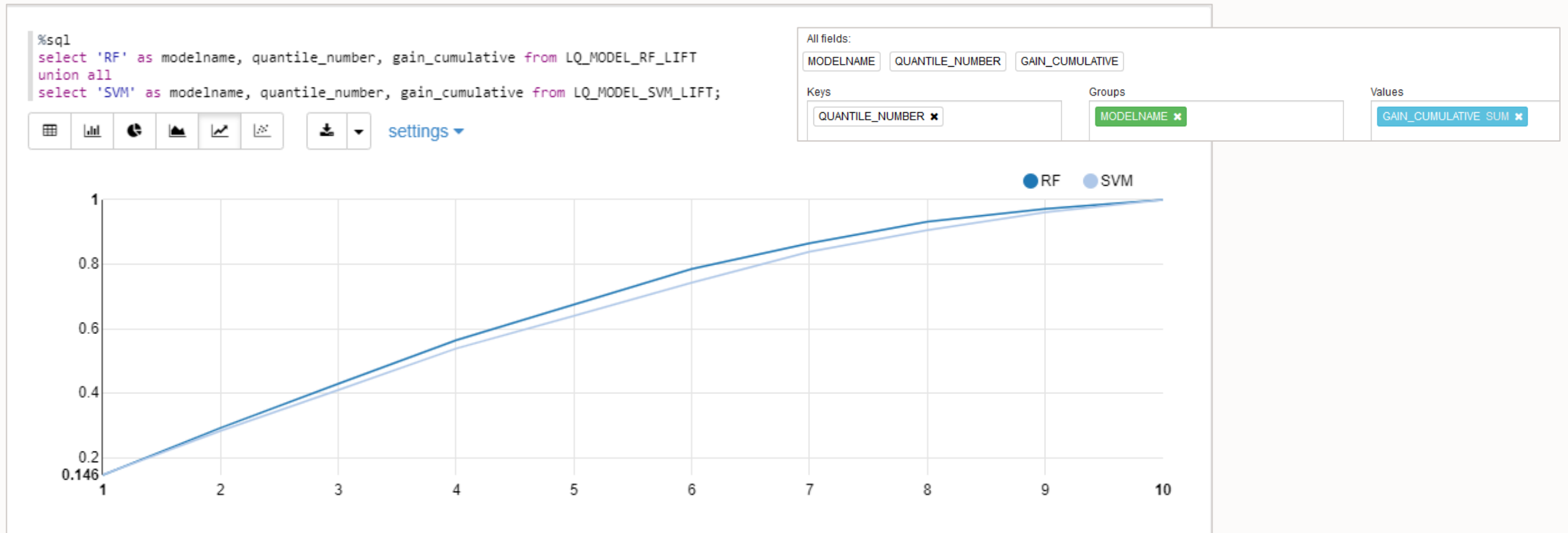
2項分類の場合にのみ使用

ポジティブクラスを指定してdbms_data_mining.compute_lift を実行する



作成したモデルを評価する

リフト（2つのモデルを比較する）



作成したモデルを評価する

モデル作成時の各種パラメータ設定を確認する

モデル作成時の各種パラメータ設定を確認

FINISHED ▶ ❌ 📖 ⚙️

```
%sql
SELECT model_name, setting_name, setting_value, setting_type
FROM user_mining_model_settings
where model_name like 'LQ_MODEL_%'
order by model_name, setting_name
;
```

USER_MINING_MODEL_SETTINGS から
パラメータ設定が確認できます

WHERE model_name で、モデル名を指定します

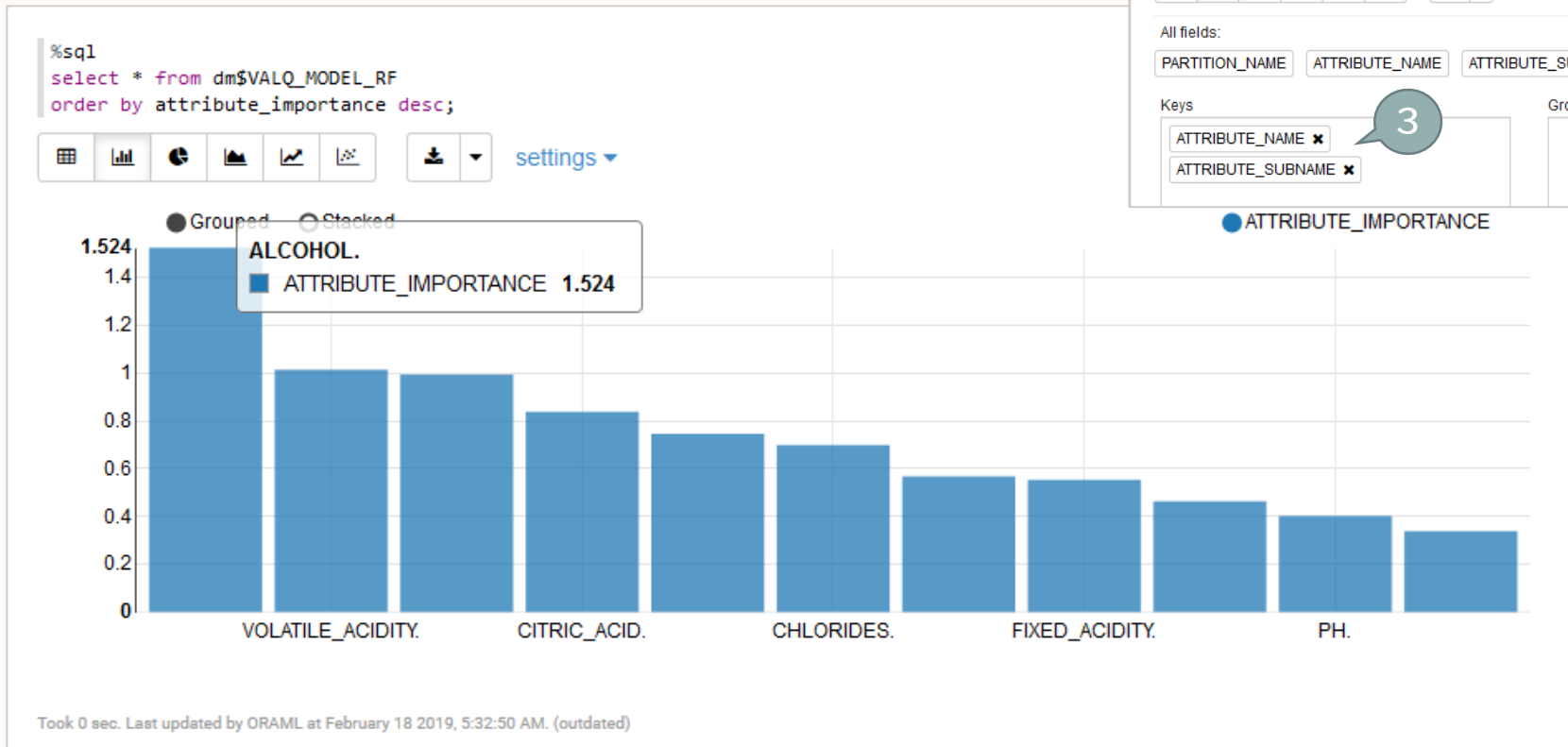
MODEL_NAME	SETTING_NAME	SETTING_VALUE	SETTING_TYPE
LQ_MODEL_RF	ALGO_NAME	ALGO_RANDOM_FOREST	INPUT
LQ_MODEL_RF	CLAS_MAX_SUP_BINS	32	DEFAULT
LQ_MODEL_RF	CLAS_WEIGHTS_BALANCED	OFF	DEFAULT
LQ_MODEL_RF	ODMS_DETAILS	ODMS_ENABLE	DEFAULT
LQ_MODEL_RF	ODMS_MISSING_VALUE_TREATMENT	ODMS_MISSING_VALUE_AUTO	DEFAULT
LQ_MODEL_RF	ODMS_RANDOM_SEED	0	DEFAULT
LQ_MODEL_RF	ODMS_SAMPLING	ODMS_SAMPLING_DISABLE	DEFAULT
LQ_MODEL_RF	PREP_AUTO	ON	INPUT



ランダムフォレスト：作成したモデルの詳細を確認

説明変数の重要度（予測したい値への影響度）を確認する

ランダムフォレストの場合は DM\$VA<モデル名>



The screenshot shows a SQL query editor interface with four numbered callouts: 1. The SQL query input field, 2. The 'settings' icon, 3. The 'ATTRIBUTE_NAME' field in the 'Keys' section, and 4. The 'ATTRIBUTE_IMPORTANCE SUM' field in the 'Values' section.

- 右のSQLを入力し実行後、
- 「棒グラフ」アイコンをクリック
 - Settingをクリック
 - XAxis, Valuesの枠内に表示したい列をドラッグ
 - X軸の値（説明変数の名前）が表示されていないものも、マウスオーバーで表示されます



SVM：作成したモデルの詳細を確認

欠損値補完や値の正規化を確認する

グローバルビュー DM\$VN<モデル名>

欠損値補完や、列値（説明変数）の正規化を
どのように行ったかがわかる

```
%sql
select * from dm$vn1q_model_svm;
```



PARTI	ATTRIBUTE_NAME	ATTRI NUMERIC_MISSING_VAL CATI	NORMALIZATION_SHIFT	NORMALIZATION_SCALE
	ALCOHOL	10.50945786682384	10.50945786682384	1.2308499387052294
	CHLORIDES	0.0459334119033592	0.0459334119033592	0.021970366607508063
	CITRIC_ACID	0.3341219799646416	0.3341219799646416	0.12046823069436083
	DENSITY	0.994008603417791	0.994008603417791	0.002905177384970001
	FIXED_ACIDITY	6.845786682380679	6.845786682380679	0.8419632282214006
	FREE_SULFUR_DIOXIDE	35.15895698291094	35.15895698291094	16.937346558165316
	PH	3.18961617950988	3.18961617950988	0.1530465389007857
	RESIDUAL_SUGAR	6.312654684737749	6.312654684737749	4.992195175234682

欠損値補完

パラメータ： ODMS_MISSING_VALUE_TREATMENT

デフォルトの挙動では、自動データ準備が使用されているかどうかにかかわらず、欠損値補完を行います

欠損値の処理が必要なアルゴリズムを使用する場合に、欠損値補完が行われます
値の平均値（カテゴリ項目の場合は最頻値）で欠損値を補完します

自動データ準備

パラメータ：PREP_AUTO

ほとんどのアルゴリズムでは、なんらかの形のデータ変換が必要となりますが、自動データ準備の設定により、モデルの作成プロセス時にアルゴリズムで必要とされる変換を自動的に実行します

決定木

自動データ準備はディビジョン・ツリーに対して効果がない、データ準備はアルゴリズムによって処理される

SVM

外れ値に敏感な正規化によって量的属性が正規化される

新規データに予測モデルを適用する

```
select
  id, -- レコードを識別したいときに利用
  prediction( LQ_MODEL_RF using *) as prediction, -- 予測
  prediction_probability( LQ_MODEL_RF , '1' using *) as probability -- 予測が 1 である確率
from <新しいデータの表>;          作成したモデル名
```

新しい液体の属性情報が
分かれば、
SQLで予測できる

つまりSQLが実行可能な
アプリから予測できます

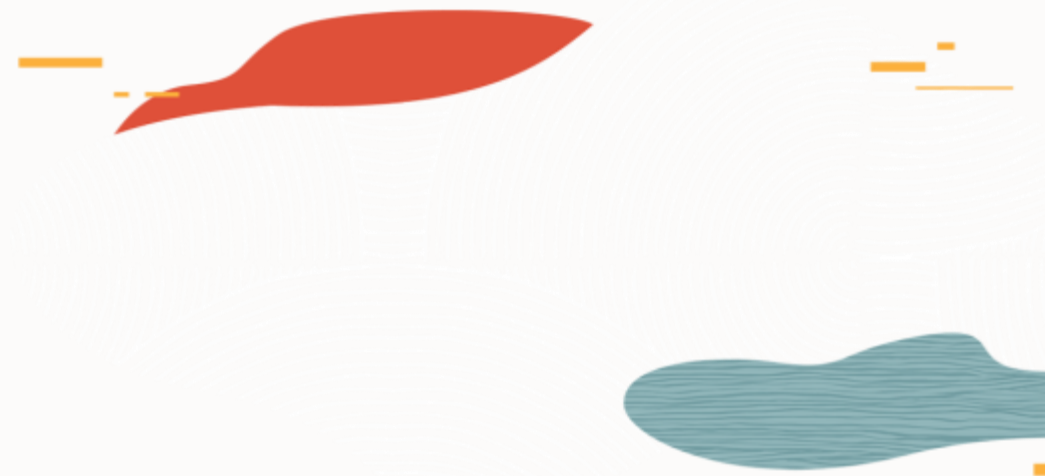
```
%sql
SELECT
  id,
  PREDICTION(LQ_MODEL_RF USING *) as prediction,
  PREDICTION_PROBABILITY(LQ_MODEL_RF, 1 USING *) as probability
FROM LQ_TEST;
```



ID	PREDICTION	PROBABILITY
2	1	0.9970588235294118
3	1	0.9269272445820432
4	0	0.3684761643585173
9	0	0.1779378370167844

新規データの代わりに、
LQ_TEST表を使って実行しています

ハンズオン②：アソシエーション分析



アソシエーション分析



豚肉を買う人は
白菜も一緒に買うこと
が多いなあ・・・



アソシエーション分析 概要

- データの中から関連が強い事象の組合せ（アソシエーション・ルール）を抽出する分析
- POSデータ（レジ精算データ）から「どの商品とどの商品が一緒に買われるか」という法則性（アソシエーションルール）を調べる目的で、米IBM社が開発した手法
- Apriori (アプリアオリ) は、アソシエーションルールを効率的に見つけ出すアルゴリズムの名称
- 同時に購入されやすい商品を見つけるマーケットバスケット分析がよく知られるが、**「Aが起きればBも起きる」** 場面に広く適用されている

アソシエーション分析 活用例

業種／業態	分析	活用により得られる効果（例）
総合スーパーマーケット コンビニエンスストア等 流通小売業全般	同時購入商品の組み合わせ 上記の店舗間比較	店舗内での商品陳列の最適化、購買動機の把握、店舗間での顧客嗜好の比較、新規出店時の地理的特徴の把握
Eコマース・デジタルコンテンツ産業 テレビショッピングチャンネル等	IDに紐づく過去の累積 購買履歴 サイト内のページ閲覧履歴	顧客ごとにカスタマイズされたメールマガジン、インターネット広告、トップページによるコンバージョン率の向上
通信サービス ITサービス 工業製品 通信業、製造業全般	製品本体とオプション製品／サービスの組み合わせ オプション製品／サービス同士の組み合わせ	携帯電話、プロバイダの料金プラン、不可オプションサービスの提案、オプション商品同士のセット販売による顧客インセンティブ
外食産業	トッピングメニューの組み合わせ 食べ物、飲み物、サイドメニューの組み合わせ	追加オーダーのリコメンド、新商品、セットメニューの考案、（時系列比較による）顧客動向の調査
銀行、証券会社、保険会社等 金融全般	金融商品の購入状況	投資対象銘柄の提案、顧客の嗜好、選択基準の把握

※参考 データアナリティクス実践講座 アクセンチュアアナリティクス著 工藤卓哉、保科学世監修



アソシエーション分析 評価指標

「A → B」：Aを買うときにBも購入する

Support (支持度)

大きいほど出現率が高い

$$\text{Support (A} \rightarrow \text{B)} = \frac{\text{A} \rightarrow \text{Bのトランザクション数}}{\text{総トランザクション数}}$$

Confidence (確信度)

大きいほどAとBの関連が強い (A購入時に、Bも一緒に購入する割合が高い)

$$\text{Confidence (A} \rightarrow \text{B)} = \frac{\text{A} \rightarrow \text{Bのトランザクション数}}{\text{A購入のトランザクション数}}$$

Lift (リフト値)

Aを購入することで、Bの購入をどれだけ促進できるか？

$$\text{Lift (A} \rightarrow \text{B)} = \frac{\text{Confidence (A} \rightarrow \text{B)} = \text{A購入時にBも買う確率}}{\text{全体のうちB購入の確率}}$$

使用するデータ



ORDER_ITEMS表の内容（レコード数：29,363）

ORDER_IDが同じ＝同時に購入した製品であることを示す

- 予測モデル作成時にCASE_IDとして指定する

予測したい値

- 今回はCATEGORY列でアソシエーション分析を行う
- 設定表に指定する



ORDER_ID	CATEGORY (カテゴリー)	ITEM (商品名称)	CNT
O_00001	ミネラルウォーター	富士山のおいしい水1L	1
O_00001	卵	卵6個パック	1
O_00001	牛乳	搾りたて牛乳 1 L	1
O_00002	ミネラルウォーター	特選水 500ml	1
O_00002	ヨーグルト	ヨーグルトいちご味	1

※ORDER_ID, CATEGORY以外の列は今回のアソシエーション分析では使用しません



Autonomous Data Warehouseにデータロード (CSVファイルをObject Storage経由でロードする場合)

Oracle Database



expダンプファイル
または
CSV等テキスト形式

Oracle以外の
システム



① ファイルを
Object Storageに置く

Autonomous Data Warehouse



② 認証情報を設定し
DBMS_CLOUDパッケージを
実行してロード

Object Storage上の
バケット

クラウド環境

Autonomous Data Warehouseにデータロード (CSVファイルをObject Storage経由でロードする場合)

認証情報の設定

(OBJ_STORE_CREDという名前のクレデンシャルを作成している)

```
%script
begin
  DBMS_CLOUD.create_credential (
    credential_name => 'OBJ_STORE_CRED',
    username => 'Cloud上のユーザ名',
    password => '指定のパスワード' } Cloudコンソール上で別途登録
  );
end;
/
```

DBMS_CLOUD.COPY_DATAでCSVファイルからロード

```
begin
  dbms_cloud.copy_data(
    table_name => 'ロード先のテーブル名',
    credential_name => 'OBJ_STORE_CRED',
    file_uri_list => 'https://xxxxxx.region-xxx.oraclecloud.com/n/<namespace>/b/バケット名/o/ファイル名',
    format => json_object('type' value 'csv', 'ignoremissingcolumns' value 'true')
  );
end;
/
```

詳細ご参考：Autonomous Data Warehouse Cloud 様々なデータロードガイド

<https://speakerdeck.com/oracle4engineer/autonomous-data-warehouse-cloud-yang-nadetarodogaido>

データを確認する 件数などの基本的な情報

ハンズオン②用に
新しいノートブックを
作成してから
実施するとよいです

ノートブックの作成 ×

名前
アソシエーション分析

コメント

接続
Global

OK 取消

- 1 パラグラフ内の先頭行に`%sql`、次の行に実行したいSQLを入力し実行

```
%sql  
select count(*) from order_items;
```

1

2

COUNT(*)
29363

- 2 実行結果が表示されます

データを確認する

探索的データ分析（SQLで実行する）

一度の買物で購入する商品個数は？

```
%sql  
select order_id, count(*) from order_items group by order_id order by count(*) desc;
```



ORDER_ID	COUNT(*)
o_10001	20
o_11112	
o_11121	
o_11624	

最も多く買われたカテゴリは？

```
%sql  
select category, count(*) from order_items group by category order by count(*) desc;
```



CATEGORY	COUNT(*)
ミネラルウォーター	1788
卵	1348
スパゲッティ	1306
フライドポテト	1282

一意なORDER_IDの数は？（何回分の購入か？）

```
%sql  
select count(distinct order_id) from order_items;
```



COUNT(DISTINCTORDER_ID)

7501

FINISHED

データを確認する 最も多く買われたカテゴリは？

「CATEGORY」をクリック

このあたりにマウスを置いて右側にドラッグ

このあたりにマウスを置いてメニューを表示し、「ソート」>「CATEGORY/CNT高から低」を選択

Ctrlキーを押下しながら「CNT」をクリック

Ctrlキーを押下し選択することで、複数選択

1

2

3

4

119 種

CATEGORY	CNT
ネラルウォ...	1.8K
スバグッティ	1.5K
チョコレート	1.2K
ミルク	1.0K
冷凍野菜	0.9K
ハンバーガー	0.8K
クッキー	0.7K
低脂肪ヨーグ...	0.6K
トマト	0.5K
フローズンス...	0.4K
チキン	0.3K
粉チーズ	0.2K
スープ	0.1K
はちみつ	0.1K
焼きたてのパ...	0.1K
ブラウニー	0.1K
ホットドッグ	0.1K
トマトジュース	0.1K
全粒粉パンタ...	0.1K
ヨーグルトク...	0.1K
エネルギーバー	0.1K
栄養ドリンク	0.1K
コーンフレーク	0.1K
マフィン	0.1K
フランスワイン	0.1K
毎	0.1K
アーモンド	0.1K
キノコのクリ...	0.1K
プロテインバー	0.1K
白ワイン	0.1K
ライトクリーム	0.1K
紅茶	0.1K
フロマージュ...	0.1K
茄子	0.1K
濃厚ダークチ...	0.1K
ヤマモ	0.1K
バーベキュー...	0.1K
低脂肪牛乳	0.1K
ズッキーニ	0.1K
ブルーベリー	0.1K
亜麻の種子	0.1K
抗酸化ジュース	0.1K
ベーコン	0.1K
服のアクセサ...	0.1K
エシヤロット	0.1K
ほうれん草	0.1K
ペットフード	0.1K
炭酸水	0.1K
マヨネーズ	0.1K
バーガーソース	0.1K
ハンディサイ...	0.1K
キャンパー	0.1K
カリフラワー	0.1K
サンドイッチ	0.1K
オートミール	0.1K
デザートワイン	0.1K
チャツネ	0.1K
お茶	0.1K
クリーム	0.1K
水スプレー	0.1K

データを確認する

一度の買物で購入する商品個数は？

「ORDER_ID」をクリック

2

このあたりにマウスを置いて右側にドラッグ

3

4

このあたりにマウスを置いてメニューを出し「ソート」>「ORDER_ID/CNT高から低」を選択

Ctrlキーを押下しながら「CNT」をクリック

1 「+」をクリックしキャンバスを追加

キャンバス1 ▼ キャンバス

7501 棒

ORDER_ID	CNT
o_10001	24
o_17495	20
o_11726	16
o_11380	12
o_15052	10
o_11822	8
o_14796	7
o_10917	6
o_13375	5
o_16462	4
o_11306	4
o_13686	4
o_15720	4
o_10302	4
o_11782	4
o_13517	4
o_15381	4
o_16894	4
o_10839	4
o_12028	4
o_13498	4
o_14914	4
o_16164	4
o_17482	4
o_11088	4
o_12103	4
o_13281	4
o_14338	4
o_15443	4
o_16376	4
o_17306	4
o_10633	4
o_11451	4
o_12161	4
o_13025	4
o_13937	4
o_14617	4
o_15415	4
o_16252	4
o_17039	4
o_10253	4
o_10951	4
o_11594	4
o_12329	4
o_12847	4
o_13366	4
o_14016	4
o_14600	4
o_15293	4
o_15901	4
o_16490	4
o_17047	4



データを確認する 一意なORDER_IDの数は？

1 「マイ計算」を右クリック

2 このように入力

3 検証

4 保存

名前: オーダー件数
f(x)
COUNT(DISTINCT ORDER_ID)

5 「+」をクリックしキャンバスを追加

6 「オーダー件数」を右側にドラッグ

オーダー件数

7,501

モデル生成に使用するオブジェクト 本ハンズオンで使用するオブジェクトの名前

Oracle Advanced Analyticsを使用した機械学習では、使用するアルゴリズムのパラメータ設定や、結果の評価データ格納先として、テーブルを使用します

テーブル名やモデル名は任意ですが（実行時に指定する）、ハンズオンでは以下の名前で作成します
同じ名前で再作成する場合は、事前にこれらを削除（テーブルの削除と、モデルの削除）します

作成するモデルの名前： **AR_MODEL**

	テーブルの名前
モデルの設定（パラメータ格納）	AR_MODEL_SETTING
結果を格納	AR_MODEL_RESULT

※2回目以降に同じ名前で実行する場合

アソシエーション分析：初期化

```
%script  
CALL DBMS_DATA_MINING.DROP_MODEL('AR_MODEL');  
DROP Table AR_MODEL_SETTING PURGE;  
DROP TABLE AR_MODEL_RESULT PURGE;
```



モデル作成

モデル作成で使用するパラメータの設定

- 1つのパラグラフ内に、以下を入力し、実行します。パラグラフのタイトルを入力します。

使用するパラメータの設定

```
%script  
CREATE TABLE ar_model_setting (  
  setting_name VARCHAR2(30),  
  setting_value VARCHAR2(4000));
```

パラメータ格納表

パラメータ設定

```
begin  
  INSERT INTO ar_model_setting VALUES (dbms_data_mining.asso_min_support,0.01);  
  INSERT INTO ar_model_setting VALUES (dbms_data_mining.asso_min_confidence,0.01);  
  INSERT INTO ar_model_setting VALUES (dbms_data_mining.asso_max_rule_length,3);  
  INSERT INTO ar_model_setting VALUES (dbms_data_mining.odms_item_id_column_name, 'CATEGORY');  
  commit;  
end;  
/
```

ASSO_MAX_RULE_LENGTHに3を設定

ルールの最大長に3を指定すると

A, B => C (AとB購入時にCを購入する)

A => C (A購入時にCを購入する)

のように、最大で2種類から1つとなるようなルールを出す

※右側(=C)は常に1つ

ODMS_ITEM_ID_COLUMN_NAMEに列名 CATEGORYを設定

※ ここから先の作業、シングルコーテーション ' で囲まれる箇所では、
全て大文字で入力することを推奨します (一部の設定項目は大文字小文字の違いを認識するため)

モデル作成（アソシエーション分析を実行）

ORDER_ITEMS表とORDER_ID列を指定して実行

2 次の新しいパラグラフ内に、以下を入力し、実行します。

アソシエーション分析を実行

```
%script
BEGIN
  DBMS_DATA_MINING.CREATE_MODEL(
    model_name           => 'AR_MODEL',
    mining_function      => DBMS_DATA_MINING.ASSOCIATION,
    data_table_name      => 'ORDER_ITEMS',
    case_id_column_name  => 'ORDER_ID',
    settings_table_name  => 'AR_MODEL_SETTING'
  );
END;
/
```

モデル名

マイニング機能の名前

DBMS_DATA_MINING.ASSOCIATION：を指定

使用するデータのテーブル名

ケースID

パラメータ格納表



モデル作成時のパラメータを確認

モデル作成時の各種パラメータ設定を確認

```
%sql
SELECT setting_name, setting_value, setting_type
  FROM user_mining_model_settings
 WHERE model_name = 'AR_MODEL'
 ORDER BY setting_name;
```

パラメータ名	設定値	ユーザ指定 or デフォルト
SETTING_NAME	SETTING_VALUE	SETTING_TYPE
ASSO_MAX_RULE_LENGTH	3	INPUT
ASSO_MIN_CONFIDENCE	.01	INPUT
ASSO_MIN_REV_CONFIDENCE	0	DEFAULT
ASSO_MIN_SUPPORT	.01	INPUT
ASSO_MIN_SUPPORT_INT	1	DEFAULT

USER_MINING_MODEL_SETTINGS から
パラメータ設定が確認できます

WHERE model_name で、モデル名を指定します

アソシエーション分析（結果を確認）

DM\$VA<モデル名> ビューを確認

NUMBER_OF_ITEMSは、ルール内の品物の数
A→Cのとき2，A,B →C のとき3になる

```
%sql
select rule_id, ANTECEDENT_PREDICATE, CONSEQUENT_PREDICATE, RULE_SUPPORT, RULE_CONFIDENCE, RULE_LIFT, number_of_items
from DM$VAAR_MODEL order by rule_lift desc, rule_id;
```

RULE_ID	ANTECEDENT_PREDICATE	CONSEQUENT_PREDICATE	RULE_SUPPORT	RULE_CONFIDENCE	RULE_LIFT	NUMBER_OF_ITEMS
225	牛ひき肉	ハーブ&ペッパー	0.015997866951073192	0.16282225237449119	3.2919938411349281	2
226	ハーブ&ペッパー	牛ひき肉	0.015997866951073192	0.32345013477088946	3.2919938411349281	2
360	スパゲッティ	牛ひき肉	0.017064391414478068	0.2857142857142857	2.9079278930025199	3
360	ミネラルウォーター	牛ひき肉	0.017064391414478068	0.2857142857142857	2.9079278930025199	3
331	スパゲッティ	オリーブオイル	0.0102652927960271964	0.171875	2.6007861842105265	2
331	ミネラルウォーター	オリーブオイル	0.0102652927960271964	0.171875	2.6007861842105265	2
195	冷凍野菜	トマト	0.016131182508998799	0.16923076923076924	2.4744639376218318	2

RULE_IDが同じものは一連でひとつのルール
RULE_ID=360 は、「スパゲッティ と ミネラルウォーター」を買う→「牛ひき肉」を買う



結果の活用例

相関が高いルールを施策に役立てる

LIFT値>1のルールを
リコメンデーション用の
ルールとして採用する

```
%sql  
select rule_id, ANTECEDENT_PREDICATE, CONSEQUENT_PREDICATE, RULE_SUPPORT, RULE_CONFIDENCE, RULE_LIFT, number_of_items  
from DM$VAAR_MODEL order by rule_lift desc, rule_id;
```



RULE_ID	ANTECEDENT_PREDICATE	CONSEQUENT_PREDICATE	RULE_SUPPORT	RULE_CONFIDENCE	RULE_LIFT	NUMBER
225	牛ひき肉	ハーブ&ペッパー	0.015997866951073192	0.16282225237449119	3.2919938411349281	2
226	ハーブ&ペッパー	牛ひき肉	0.015997866951073192	0.32345013477088946	3.2919938411349281	2
360	スパゲッティ	牛ひき肉	0.017064391414478068	0.2857142857142857	2.9079278930025199	3
360	ミネラルウォーター	牛ひき肉	0.017064391414478068	0.2857142857142857	2.9079278930025199	3
224	スパゲッティ	オリーブオイル	0.010265297960271964	0.171875	2.6097861842105265	3
	オリーブ		0.010265297960271964	0.171875	2.6097861842105265	3
			0.2508998799	0.16923076923076924	2.4744639376218318	2

買い物かごに「牛ひき肉」
が入ったら、
「ハーブ&ペッパー」を
リコメンドする

「牛ひき肉」棚の近くに
ハーブ&ペッパーを
配置する



参考：結果の確認（ルールを1行で表現）

まずは同一RULE_ID内での通し番号を付与(piece列)

アソシエーション分析 結果の確認（ルールを1行で表現）

```
%script
create or replace view ar_result_tmp as
select rule_id,
ANTECEDENT_PREDICATE, CONSEQUENT_PREDICATE, RULE_SUPPORT,
RULE_CONFIDENCE,
RULE_LIFT,
number_of_items,
row_number() over (partition by rule_id order by ANTECEDENT_PREDICATE desc) piece
from DM$VAAR_MODEL;
```


参考：結果の確認（ルールを1行で表現）

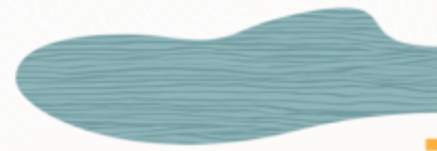
```
create table AR_MODEL_RESULT as
select
  r1.rule_id,
  --r1.ANTECEDENT_PREDICATE,
  --r2.ANTECEDENT_PREDICATE,
  case
    when r2.ANTECEDENT_PREDICATE is not null then r1.ANTECEDENT_PREDICATE || ', ' || r2.ANTECEDENT_PREDICATE
    else r1.ANTECEDENT_PREDICATE
  END as ANTECEDENT_PREDICATE_ALL,
  r1.CONSEQUENT_PREDICATE,
  r1.RULE_SUPPORT,
  r1.RULE_CONFIDENCE,
  r1.RULE_LIFT,
  r1.number_of_items
from
  (
    select * from ar_result_tmp where piece = 1
  ) r1
left outer join
  (
    select * from ar_result_tmp where piece = 2
  ) r2
on r1.rule_id = r2.rule_id
;
```

アソシエーション分析 結果の確認（ルールを1行で表現）

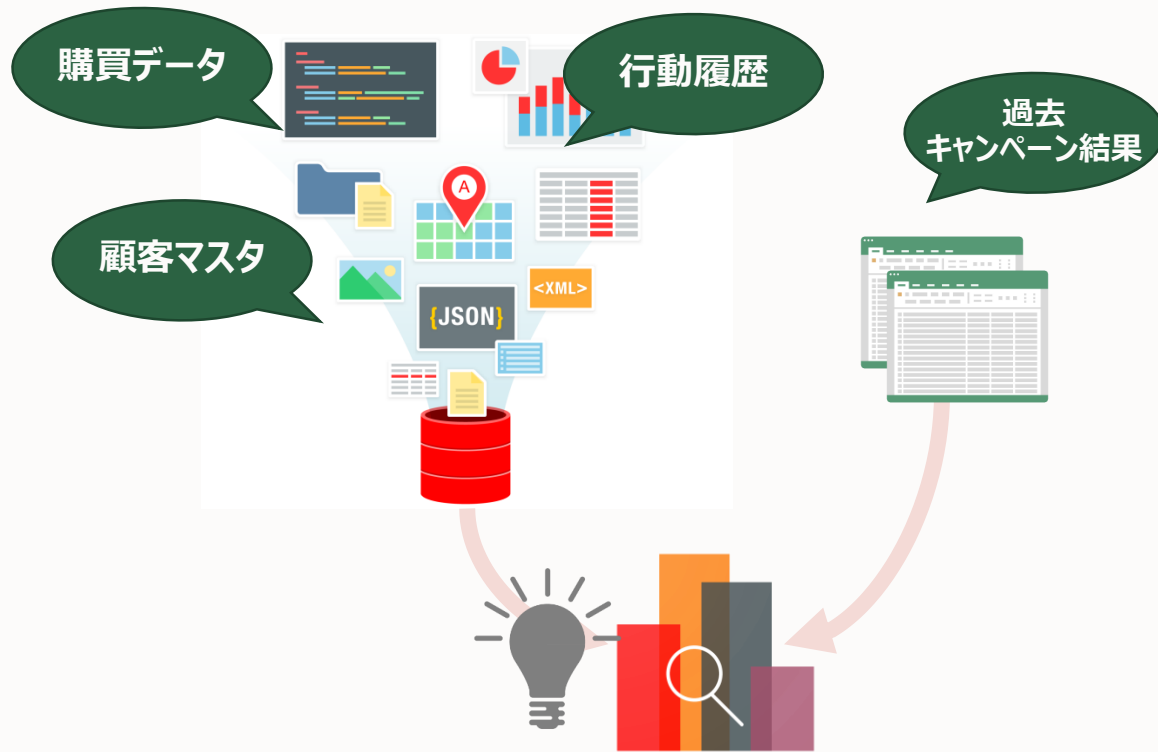
```
%sql
SELECT * from AR_MODEL_RESULT order by rule_lift desc, rule_confidence desc;
```

RULE_ID	ANTECEDENT_PREDICATE_ALL	CONSEQUENT_PREDICATE	RULE_SUPPORT	RULE_C
226	ハーブ&ペッパー	牛ひき肉	0.015997866951073192	0.323450
225	牛ひき肉	ハーブ&ペッパー	0.015997866951073192	0.162822
360	ミネラルウォーター, スパゲッティ	牛ひき肉	0.017064391414478068	0.285714
331	ミネラルウォーター, スパゲッティ	オリーブオイル	0.010265297960271964	0.171875
196	トマト	冷凍野菜	0.016131182508998799	0.235867
195	冷凍野菜	トマト	0.016131182508998799	0.169230

まとめ：Autonomous Data Warehouseではじめるデータ活用基盤



データがあれば x 分析ができれば データ活用で実現するデータドリブンな世界



マーケティング業務でのデータ活用例



働き方改革でのデータ活用例



データ活用をスタートしたい とはいえ、すぐにスタートできない現実的な問題

環境構築に
時間を
かけたくない

データ活用は
したいけど
運用が心配

パフォーマンス
も心配

費用が気に
なる・・・

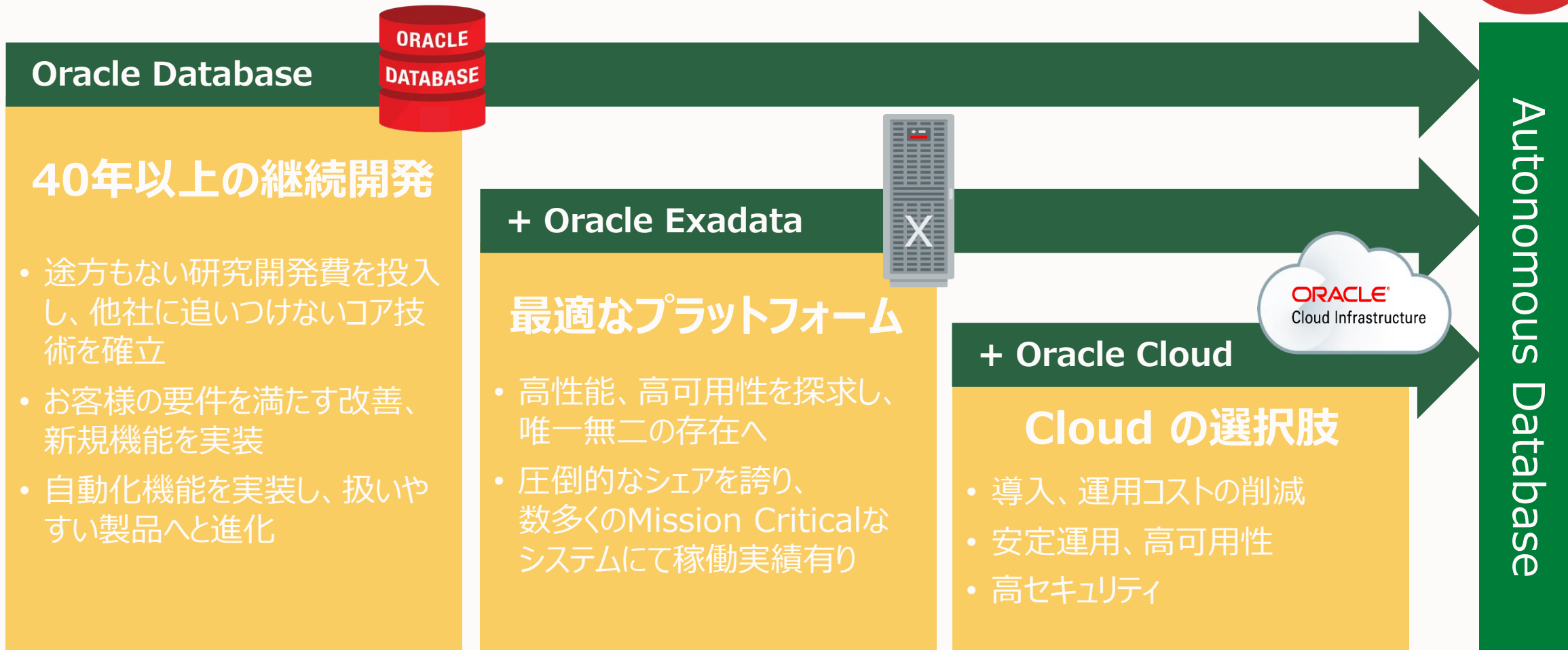
セキュリティは
大丈夫？



**Autonomous Data Warehouse
で解決**

Autonomous Database までの進化

これまでの全ての開発/進化はAutonomous Databaseの実現へ続いてきた



Autonomous Databaseの作成 簡単！名前やサイズなどを入力するだけ！

ORACLE Cloud

- コア・インフラストラクチャ
- コンピュート
- ブロック・ストレージ
- オブジェクト・ストレージ
- ファイル・ストレージ
- ネットワーキング
- データベース
- ベア・メタル、VMおよびExadata
- Autonomous Data Warehouse**
- Autonomous Transaction Processing
- ソリューションおよびプラットフォーム
- 分析
- リソース・マネージャ
- 電子メール配信

Autonomous Database

- Autonomous Database**
- Autonomous Container Database
- Autonomous Exadata Infrastructure

Autonomous Databaseの作成

名前	データベース名	状態
----	---------	----

「Autonomous Databaseの作成」
をクリック

ORACLE Cloud

Autonomous Databaseの作成

Autonomous Databaseの基本情報の指定

コンパartmentの選択
SE
oraclejapan (リージョン)SE

表示名
oramyadw

データベース名
oramyadw

名前、英字と数字のみを含めて、英字で始める必要があります。最大14文字です。

ワークロード・タイプの選択

データ・ウェアハウス
大規模データのスキャン操作に重点を置き
またはデータ・ウェアハウス・ワークロー
スが構成されます。

トランザクション処理

ワークロードの種類
(Autonomous Data Warehouseの
場合はデータウェアハウスを選択)

データベースを構成します

CPUコア数
1

ストレージ(TB)
1

ストレージの量。

ワークロードの増加に応じて、プロビジョニングコア数の9倍まで使用することをシステムに許可します。[詳細](#)

管理者資格証明の作成 ①

ユーザー名 読取り専用
ADMIN

パスワード
.....

パスワードの確認
.....

管理者パスワード

ライセンス・タイプの選択

ライセンス持込み

自
ン
ア
行しま (詳細)

ライセンス込み
新しいデータベース・ソフトウェア・ライセンスとデー
タベース・クラウド・サービスをサブスクライブします。

Show Advanced Options

Autonomous Databaseの作成

使用条件およびプライバシー クッキー設定



Autonomous Database が実現すること

完全な自動化によるデータ管理の効率化、コストの削減

オンプレミス

- AP管理
- AP最適化
- DB最適化/スケール
- HA/DR
- バックアップ/リストア
- DBインストール/パッチ適用
- OSインストール/パッチ適用
- サーバー管理
- ファシリティ管理

全て
お客様管理

Autonomous Database

- AP管理
- AP最適化
- DB最適化/スケール
- HA/DR
- バックアップ/リストア
- DBインストール/パッチ適用
- OSインストール/パッチ適用
- サーバー管理
- ファシリティ管理

お客様管理

オラクル管理

お客様は
データ活用に
集中できる

管理コストを最大80%、TCOを3年間で最大50%削減

*Source: Oracle TCO report 2018. 50% savings calculated based on 16 CPU config w/BYOL from on-premises Oracle Database



Autonomous Data Warehouse お客様事例

コストを削減し、顧客サービス向上や顧客向けのリワードへ投資



Hertz

安全性の高い環境を人手をかけずに
Creates a Highly Secure Labor Free Environment

2-3
手作業での
週次パッチ当てが
なしに

投資
変革に向け

安全に
手作業による運用面でのリスクをなしに



Autonomous Data Warehouse お客様事例

財務レポートの効率化による業務改善、データドリブンな意思決定の実現



JAMES ANTHONY

CTO, DATA INTENSITY

<https://www.youtube.com/watch?v=4TCJLhbzRFU>

• 低コストで業務効率改善

以前は財務データを抽出するだけで、時間の60%を費やしていたが、ADWは、決算期などの繁忙期にはリソースを拡張することで十分な性能が得られる
(残りはスケールダウンしほとんど費用が発生しない)

• データドリブンの実現

CFO自ら役員会でその場でデータ分析
従来の10倍のメンバーが分析に関与するように

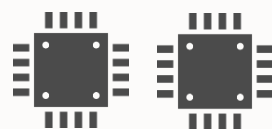
• DB管理不要

DBはクラウド上でOracleが管理しているため、IT部門の負担がない

再掲：機械学習による予測

INPUT

大量のデータ



情報処理

データの塊から
何かを予測する
予測モデルを作成

OUTPUT(機能=できること)

新しいデータに対し
予測モデルを適用して
活用する

データの準備

予測モデルを
作成・評価

実装
(サービスへの
組み込み)

データの準備、検索、管理にかかる時間が
8割を占めています。

80%

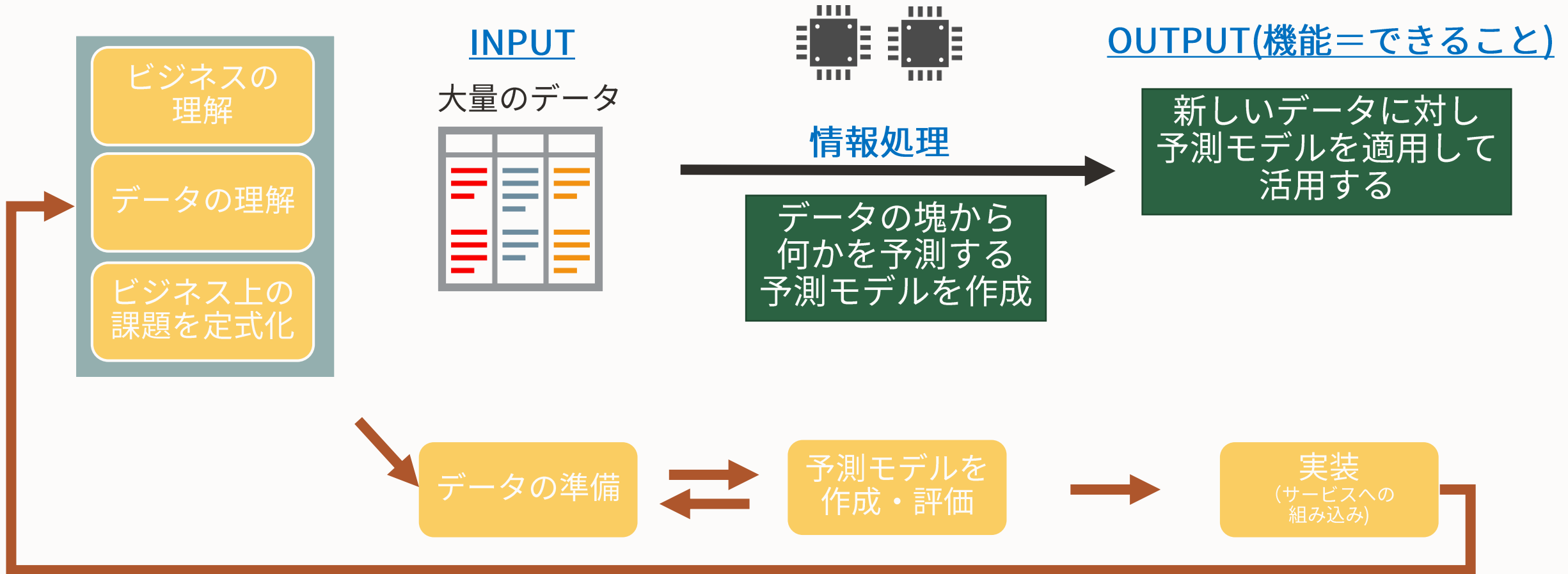
of time is still spent preparing, searching,
and governing data



IDC

Data Literacy: A Foundation for Succeeding in a Data-Driven World

機械学習プロジェクト全体



まとめ

Oracle Machine Learning を使用した機械学習

Webブラウザベース、Notebook 上でのSQL実行

Oracle Machine Learning を使って In-Database Analytics

機械学習は試行錯誤を繰り返すプロジェクトであり、それに適する環境が望ましい

Databaseのパワーを活かした簡単・高速なデータ加工処理

データのある場所で、データを移動することなく、予測モデルを作成

製品ドキュメント

Autonomous Data Warehouse

https://docs.oracle.com/cd/E83857_01/paas/autonomous-data-warehouse-cloud/books.html

Oracle Machine Learningの使用

Oracle Database 19c

https://docs.oracle.com/cd/F19136_01/books.html

Data Mining概要

Data Miningユーザーズ・ガイド

Data Mining APIガイド

Database PL/SQLパッケージおよびタイプ・リファレンス



ORACLE