

日々はOracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2020年4月2日 木曜日

Oracle Machine Learningの機能をOracle APEXで使用する

[Autonomous Database for APEX Developers](#)のプログラムに含まれている[Lab 300: Adding Machine Learning to your APEX application](#)を試してみました。それで、どのような作業になったかを紹介します。Always Freeとして提供されているAutonomous Databaseを使用して演習を進めます。

準備

以下の3つの作業が完了していれば、準備は完了です。

1. Oracle Cloudのアカウント取得 - [無料Oracle Cloudプロモーションへのサインアップ](#)
2. (Always Freeの)Autonomous Databaseのインスタンス作成 - [こちらの記事](#)の前半部分
3. Oracle APEXのワークスペース作成 - [同じ記事](#)の後半部分

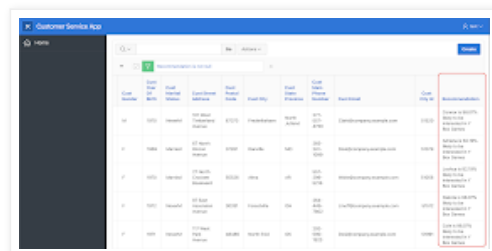
作業自体は[Oracle Spatialの機能をOracle APEXで使用する](#)と同じなので、すでにこの演習を行っていれば準備はできています。

Oracle Machine Learningの機能をOracle APEXのアプリケーションで使用する

この演習では、あらかじめ提供されているOracle APEXのアプリケーションをインポートし、そのアプリケーションにOracle Machine Learningが提供する機能を使った処理を追加します。

作成するアプリケーションの要件は以下になります。

1. あるゲーム・コンソールのメーカーが販売している"Y Box Games"という製品があります
2. このメーカーは（この製品に限らない）顧客リストと、その顧客のプロファイルを持っています
3. 顧客ごとの"Y Box Games"を含む各種製品の販売実績を持っています
4. まだ"Y Box Games"を購入していない顧客の中で、顧客ごとに、この製品の購買意欲がどの程度になるかを導き出します - **Oracle Machine Learning!**
5. 販売員に、顧客が"Y Box Games"に、どの程度の購買意欲を持ちそうか伝えます - **Oracle APEX!**



Customer ID	Name	Address	Product	Product Name	Product Price	Product Type	Product Category	Product Subcategory	Product Description
1	John Doe	123 Main St, New York, NY 10001	Y Box Games	Y Box Games	100.00	Game	Video Game	Console Game	Y Box Games Console Game
2	Jane Smith	456 Elm St, Los Angeles, CA 90001	Y Box Games	Y Box Games	100.00	Game	Video Game	Console Game	Y Box Games Console Game
3	Mike Johnson	789 Oak St, Chicago, IL 60601	Y Box Games	Y Box Games	100.00	Game	Video Game	Console Game	Y Box Games Console Game
4	Sarah Lee	101 Pine St, San Francisco, CA 94101	Y Box Games	Y Box Games	100.00	Game	Video Game	Console Game	Y Box Games Console Game
5	David Kim	202 Birch St, Seattle, WA 98101	Y Box Games	Y Box Games	100.00	Game	Video Game	Console Game	Y Box Games Console Game

機械学習に使用するデータは、Oracleが提供しているSHスキーマのサンプルに含まれるSUPPLEMENTARY_DEMOGRAPHICS表のデータになります。この表は以下の属性を持ちます。

- CUST_ID: 顧客番号
- EDUCATION: 最終学歴
- OCCUPATION: 職業
- HOUSEHOLD_SIZE: 世帯人数
- YRS_RESIDENCE: 居住年数

- AFFINITY_CARD: 優待カードあり
- BULK_PACK_DISCKETS: ディスク・パック - バルク品購入実績あり
- FLAT_PANEL_MONITOR: フラット・パネル・モニタ購入実績あり
- HOME_THEATER_PACKAGE: ホーム・シアター機材セット購入実績あり
- BOOKKEEPING_APPLICATION: 家計簿アプリケーション購入実績あり
- PRINTER_SUPPLIES: プリンタ用品購入実績あり
- Y_BOX_GAMES: Y Box Games購入実績あり
- OS_DOC_SET_KANJI: OS日本語ドキュメンテーションセット購入実績あり
- COMMENTS: コメント

この中のY_BOX_GAMESがターゲット属性です。ターゲット属性とその他の属性の関連を基に生成したモデルを、まだY Box Gamesを購入していない顧客に適用し、どの程度興味を持ちそうかを導出します。

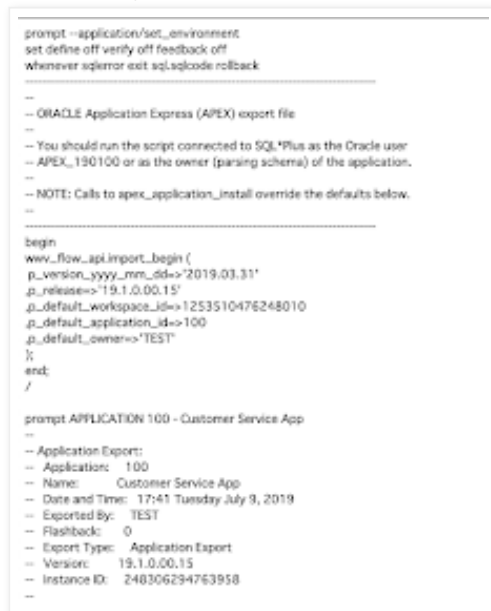
アプリケーションをインポートする

機能追加を行う元となるアプリケーションをインポートします。このアプリケーションは、メーカーのコール・センターにて、電話を受け取った販売やサポートのスタッフが顧客の情報を確認するために使用することを想定しています。

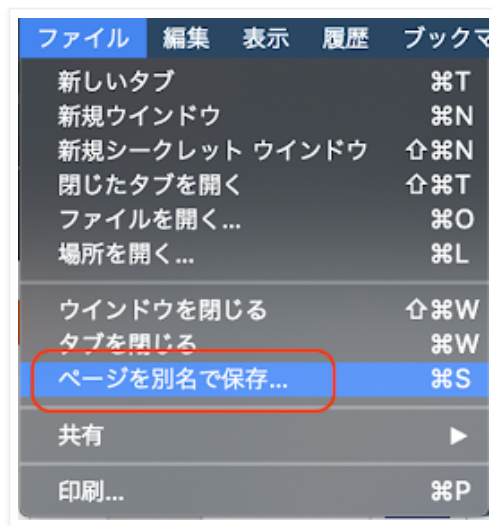
GitHubの[こちら](#)から、f100.sqlをダウンロードします。GitHubの画面上で**Raw**をクリックします。



ブラウザにファイルの内容だけが表示されます。



メニューの例はMac上のChromeですが、だいたいどのブラウザでもファイル・メニューにページを別名で保存...または別名でページを保存...といった機能はあるので、それを使ってファイルに保存します。



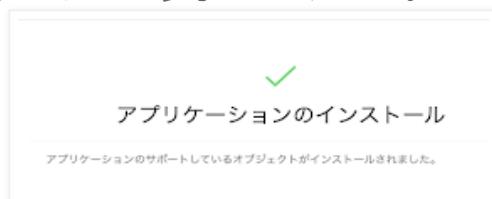
Oracle APEXのワークスペースにログインし、アプリケーション・ビルダーを開いて、いまダウンロードしたファイルをアプリケーションとしてインポートします。



インポートのファイルのインポートとして、ダウンロードしたf100.sql(またはf100.sql.txt)を指定します。ファイル・タイプはデフォルトで指定される、データベース・アプリケーション、ページまたはコンポーネントのエクスポートのままにし、次へ進みます。

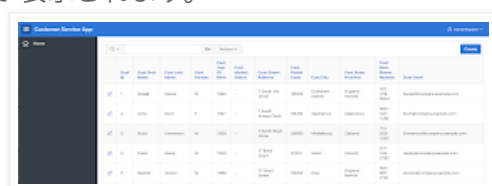


これ以降のインポート・ウィザードからの質問はすべてデフォルトを選択して進みます。最終的に以下の緑のチェックの表示を含む画面になればアプリケーションのインポートは正常終了しています。ファイルの編集を除いて、**Oracle Spatialの機能をOracle APEXで使用するのAPEXアプリケーションをインポートするとほぼ同じです**ので、より細かいステップが必要であればそちらも参考にしてください。



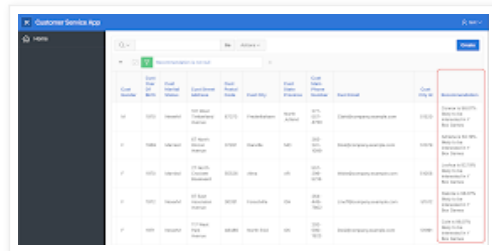
作成されたアプリケーションを確認する

インポートしたアプリケーションを実行します。ワークスペースのログイン時に与えたユーザー名、パスワードでログインすると以下の対話レポートが表示されます。



顧客の名前、性別、生年、婚姻歴、居住地、電話番号、メール・アドレスが一覧されています。この対話レポー

トの表示に、以下のRecommendationの項目を追加することが、この演習の作業になります。



ID	Category	Severity	Recommendation	Created Time	Last Updated	Accepted
1	Database	High	Upgrade to the latest version of the database software.	2023-10-27 10:00:00	2023-10-27 10:00:00	Not Accepted
2	Database	Medium	Configure the database to use the latest security patches.	2023-10-27 10:00:00	2023-10-27 10:00:00	Not Accepted
3	Database	Low	Review the database configuration for optimal performance.	2023-10-27 10:00:00	2023-10-27 10:00:00	Not Accepted
4	Database	High	Enable the database to use the latest security patches.	2023-10-27 10:00:00	2023-10-27 10:00:00	Not Accepted
5	Database	Medium	Configure the database to use the latest security patches.	2023-10-27 10:00:00	2023-10-27 10:00:00	Not Accepted

Oracle Machine Learningユーザーを作成する

機械学習のモデルの作成には、Autonomous Databaseに付属しているZeppelinを使用します。ZeppelinはWebベースのノートブック環境で、データの分析や操作を行うことができます。少々、ノートブックって？と思って調べると、[Kunuth先生](#)の提唱する文芸的プログラミングを行うために使用するもの、とのこと。コードをドキュメントに埋め込み、文章として読めて、かつ、コンピュータで実行可能とする、といった環境です。Autonomous Databaseに組み込まれているZeppelinベースのOracle Machine Learningノートブックに記載するコードは、SQLかPL/SQLになります。

Oracle Machine Learningユーザー（以後、MLユーザーとします）を作成するには、まず、Autonomous Databaseのサービス・コンソールを開きます。



サービス・コンソールの中の**管理**を選択し、**Oracle MLユーザーの管理**を開きます。



デフォルトでシステム管理者としてADMINユーザーが登録されていますが、これは使わず、新規にMLユーザー **mluser1**を作成します。**作成**をクリックします。



ユーザー名、名、姓、電子メール・アドレスを指定し、ユーザーにパスワードと電子メール・アカウントの詳細を生成してください。云々、と記載されているチェックボックスのチェックを外します。チェックを入れるとパスワードが自動再生され、設定した電子メール・アドレスに通知されます。ユーザーは初回ログイン時にパスワードを変更する必要があります。今回はチェックを外して、**パスワードは自分で設定**します。



(別のタブで開いている)サービス・コンソールに戻り、開発から**Oracle Machine Learning**ノートブックを開きます。



ノートブックへのサイン・イン画面が開くので、先ほど作成した**ユーザー名**と、そのユーザーの**パスワード**を入力して**サインイン**をクリックします。



ノートブックを作成する

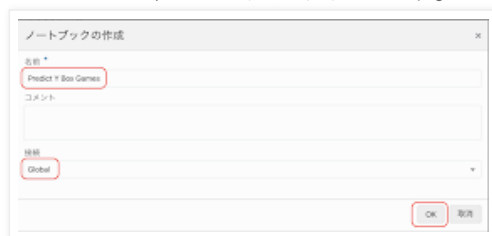
サインインするとOracle Machine Learningノートブックのホーム画面が開きます。これからの作業を行うために使用するノートブックを作成するため、**ノートブック**をクリックします。



作成済みのノートブッケー一覧が表示されます。新規のノートブックを作成するために、**作成**をクリックします。



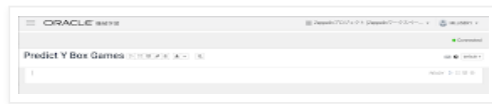
名前を**Predict Y Box Games**、接続を**Global**とし、**OK**をクリックします。



起動中の画面になります。



起動が完了すると、ノートブックの作成完了です。このノートブックには、SQL、PL/SQLの記述と実行、コメントや説明の記述を行うことができます。



機械学習モデルを構築する

Y Box Gamesをすでに購入済みの顧客の情報から、モデルを機械学習モデルを作成します。

日本語マニュアルは[こちら](#)になります。

作成するモデルの入力となるデータを、以下のSQL文を実行することで確認します。

```
SELECT * FROM SH.SUPPLEMENTARY_DEMOGRAPHICS
```

上記SQLをパラグラフの入力とし、実行します。

CUST_ID	EDUCATION	OCCUPATION	HOUSEHOLD_SIZE	YRS_RESIDENCE	ACTIVITY_CARD	BULK_PACK_DISCOUNT	PLAY_POINTS
100001	High	Other	1	0	0	1	1
100002	High	Other	1	0	0	1	1
100003	High	Other	1	0	0	1	1
100004	High	Other	1	0	0	1	1
100005	High	Other	1	0	0	1	1
100006	High	Other	1	0	0	1	1
100007	High	Other	1	0	0	1	1
100008	High	Other	1	0	0	1	1
100009	High	Other	1	0	0	1	1
100010	High	Other	1	0	0	1	1

それぞれの列についてはすでに説明済みです。検索結果のデータを眺めて、果たして、こういったプロフィールを持つ人がY Box Gamesを購入するのか、またはしたのか(Y_BOX_GAMESの値が1)、見つけることができるか考えてみましょう。

簡単には見つけられない、または、ほとんど見つけられる気がしないのではないのでしょうか。

機械学習を適用することで、ターゲットなる属性Y_BOX_GAMESとその他の属性にある関係を見つけることができます。

次の処理を記述するため、新しくパラグラフを追加します。すでにパラグラフが追加済みであれば、それをそのまま使用します。

CUST_ID	EDUCATION	OCCUPATION	HOUSEHOLD_SIZE	YRS_RESIDENCE	ACTIVITY_CARD	BULK_PACK_DISCOUNT	PLAY_POINTS
100001	High	Other	1	0	0	1	1
100002	High	Other	1	0	0	1	1
100003	High	Other	1	0	0	1	1
100004	High	Other	1	0	0	1	1
100005	High	Other	1	0	0	1	1
100006	High	Other	1	0	0	1	1
100007	High	Other	1	0	0	1	1
100008	High	Other	1	0	0	1	1
100009	High	Other	1	0	0	1	1
100010	High	Other	1	0	0	1	1

入力データを2つのセットに分割します。60%をトレーニング用、40%をテスト用とします。最初に60%のデータを含む表N1_TRAIN_DATAを作成します。

```
CREATE TABLE N1_TRAIN_DATA AS SELECT * FROM SH.SUPPLEMENTARY_DEMOGRAPHICS SAMPLE (60) SEED (1);
```

CUST_ID	EDUCATION	OCCUPATION	HOUSEHOLD_SIZE	YRS_RESIDENCE	ACTIVITY_CARD	BULK_PACK_DISCOUNT	PLAY_POINTS
100001	High	Other	1	0	0	1	1
100002	High	Other	1	0	0	1	1
100003	High	Other	1	0	0	1	1
100004	High	Other	1	0	0	1	1
100005	High	Other	1	0	0	1	1
100006	High	Other	1	0	0	1	1
100007	High	Other	1	0	0	1	1
100008	High	Other	1	0	0	1	1
100009	High	Other	1	0	0	1	1
100010	High	Other	1	0	0	1	1

次に、残りのデータを含む表N1_TEST_DATAを作成します。

```
CREATE TABLE N1_TEST_DATA AS SELECT * FROM SH.SUPPLEMENTARY_DEMOGRAPHICS MINUS SELECT * FROM N1_TRAIN_DATA;
```

CUST_ID	EDUCATION	OCCUPATION	HOUSEHOLD_SIZE	YRS_RESIDENCE	ACTIVITY_CARD	BULK_PACK_DISCOUNT	PLAY_POINTS
100001	High	Other	1	0	0	1	1
100002	High	Other	1	0	0	1	1
100003	High	Other	1	0	0	1	1
100004	High	Other	1	0	0	1	1
100005	High	Other	1	0	0	1	1
100006	High	Other	1	0	0	1	1
100007	High	Other	1	0	0	1	1
100008	High	Other	1	0	0	1	1
100009	High	Other	1	0	0	1	1
100010	High	Other	1	0	0	1	1

それぞれの表の行数を確認します。

```
select 'TRAIN', count(*) from n1_train_data
union
select 'TEST', count(*) from n1_test_data
```

TRAIN	COUNT(*)
TRAIN	60
TEST	40

機械学習(今回は決定木による分類)のモデルは、ターゲット属性(Y_BOX_GAMES)と予測子である他の属性との関係を持します。これらの関係はモデルの作成（トレーニング）プロセスにて見つけれられます。モデルを作成するには、いくつかのパラメータの指定が必要です。最初に表を作成して、これらのパラメータの指定を保存します。このパラメータを保持する表の名前は任意で指定できます。今回の演習では、使用するアルゴリズム、つまり決定木（ディジジョン・ツリー）のみを指定します。

パラメータを保存する表N1_BUILD_SETTINGSを作成します。

```
CREATE TABLE N1_BUILD_SETTINGS(SETTING_NAME VARCHAR2(30), SETTING_VALUE VARCHAR2(4000));
```



次に、アルゴリズム(ALGO_NAME)として、決定木(ALGO_DECISION_TREE)を指定します。

```
INSERT INTO N1_BUILD_SETTINGS(SETTING_NAME, SETTING_VALUE) VALUES('ALGO_NAME','ALGO_DECISION_TREE');
```



準備ができたので、モデルの作成とトレーニングを行います。以下のPL/SQLコードを実行します。最初の'%script'は、PL/SQLコードとしての実行をノートブックに指示しています（デフォルトは'sql' - SQL文の実行です。これ以外にも'md' - マークダウンの記述、といったものもあります）。

```
%script
CALL DBMS_DATA_MINING.CREATE_MODEL('N1_CLASS_MODEL', 'CLASSIFICATION',
    'N1_TRAIN_DATA', 'CUST_ID', 'Y_BOX_GAMES', 'N1_BUILD_SETTINGS');
```



DBMS_DATA_MININGパッケージ、および、CREATE_MODELプロシージャの[説明](#)はマニュアルに詳しいです。今回の指定は以下の意味になります。

1. N1_CLASS_MODEL: 作成されるモデルにつける名前です。データベースには特別な型のオブジェクトとして保存されます。
2. CLASSIFICATION: マイニング機能を指定します。今回は分類(CLASSIFICATION)を指定しています。その他に相関(ASSOCIATION)、属性評価(ATTRIBUTE_IMPORTANCE)、クラスタリング(CLUSTERING)、特徴抽出(FEATURE_EXTRACTION)、回帰(REGRESSION)、時系列(TIME_SERIES)があります。
3. N1_TRAIN_DATA: トレーニングに使用するデータを保持している表の名前です。
4. CUST_ID: トレーニング・データを一意で識別できる列名です。
5. Y_BOX_GAMES: ターゲット属性の名前です。
6. N1_BUILD_SETTINGS: モデルの作成設定が含まれる表の名前です。

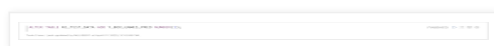
これでモデルが作成できました。

結果とモデルの精度を検証する

今回作成したモデルがどの程度の正確度で、Y Box Gamesの所有者を予測できるのかを求めてみます。この用途のために、先ほど作成したテスト用の（トレーニングに使用しなかった）データを保持している表N1_TEST_DATAを使用します。この表に含まれているデータは現実の顧客情報ですので、Y_BOX_GAMES列には予測ではない所有情報が含まれています。表N1_TEST_DATAに含まれるデータを用いてターゲット属性であるY Box Games列の結果を予測し、それと現実の情報を比較します。

予測した結果を保持する列Y_BOX_GAMES_PREDを表N1_TEST_DATAに追加します。

```
ALTER TABLE N1_TEST_DATA ADD Y_BOX_GAMES_PRED NUMBER(1);
```



次に予測を行い、結果をY_BOX_GAMES_PREDに保存します。


```
UPDATE N1_TEST_DATA SET Y_BOX_GAMES_PRED = PREDICTION(N1_CLASS_MODEL USING *);
```



SQLスコアリング関数PREDICTIONの説明は[こちら](#)になります。今回は、モデルとしてN1_CLASS_MODELを使用し、すべての予測子である属性を使用する、という指定になっています。

結果を確認します。

```
SELECT CUST_ID, Y_BOX_GAMES, Y_BOX_GAMES_PRED FROM N1_TEST_DATA;
```

CUST_ID	Y_BOX_GAMES	Y_BOX_GAMES_PRED
1000001	0	0
1000002	0	0
1000003	0	0
1000004	0	0
1000005	0	0
1000006	0	0
1000007	0	0
1000008	0	0
1000009	0	0
1000010	0	0

予測がどれほど実際の値と一致しているのか、正解率を求めてみます。

```
select to_char(
  (
    sum(case when y_box_games = y_box_games_pred then 1 else 0 end) / count(*)
  ) * 100, '999.99') correct_pred_percentage
from n1_test_data;
```

correct_pred_percentage
90.00

おおよそ90%です。

次に混同行列(confusion matrix)を求めます。ターゲットである2つの列を含めてGROUP BYすることにより、SQLにて比較的簡単に求められます。

```
SELECT Y_BOX_GAMES, Y_BOX_GAMES_PRED, COUNT(*)
FROM N1_TEST_DATA GROUP BY Y_BOX_GAMES, Y_BOX_GAMES_PRED ORDER BY 1,2;
```

Y_BOX_GAMES	Y_BOX_GAMES_PRED	COUNT(*)
0	0	1180
0	1	80
1	0	110
1	1	407

上の行から、真陰性(True Negative)、偽陽性(False Positive)、偽陰性(False Negative)、真陽性(True Positive)です。偽陰性の件数が高そうです。実際には購入しているのに購入しない、と予測していますので、実際の運用場面では、売り込みをしない機会損失になるでしょう。偽陽性はそれほどでもないのに、無駄になる売り込みはあまり発生しないようです。

予測を実行する

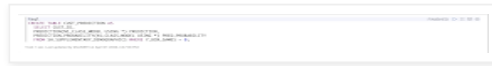
モデルの作成と検証ができました。これから予測を実行する方法には、おおよそ以下の2つの方法があります。

- ある時点ですべての顧客について、バッチで予測処理を行います。
- 予測結果が必要なときに、顧客ごとに予測処理を行います。例えば、顧客情報のフォームを開いたときなどです。

今回の演習ではバッチ処理として、予測処理を行います。

まだY Box Gamesを所有していない顧客を対象として、購入意思を持ちそうかどうかを予測します。

```
CREATE TABLE CUST_PREDICTION AS
SELECT CUST_ID,
PREDICTION(N1_CLASS_MODEL USING *) PREDICTION,
PREDICTION_PROBABILITY(N1_CLASS_MODEL USING *) PRED_PROBABILITY
FROM SH.SUPPLEMENTARY_DEMOGRAPHICS WHERE Y_BOX_GAMES = 0;
```

今後、バッチ処理にて予測を更新するには、上記処理をジョブとして登録し繰り返し実行させる必要がありますが、その実装は今回の演習には含んでいません。

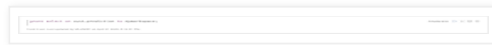
APEXアプリケーションに組み込む

インポート済みのAPEXアプリケーションに、Oracle Machine Learningによる予測結果を追加します。予測結果は対話レポートの追加の列として表示されます。

Oracle Machine Learningノートブックが所有しているスキーマに含まれている表に、Oracle APEXからアクセスできるように権限を与えます。スキーマ名はOracle APEX側で使用しているスキーマ名ですが、ワークスペースを作成するときに同時に作成したスキーマの名前になります。大抵の場合はワークスペース名と同じ名前を使用していると思います。

grant select on cust_prediction to スキーマ名;

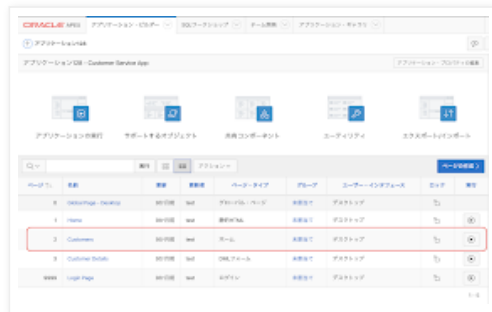
以下は、Oracle APEXのスキーマ名がMYWORKSPACEとして実行した結果になります。



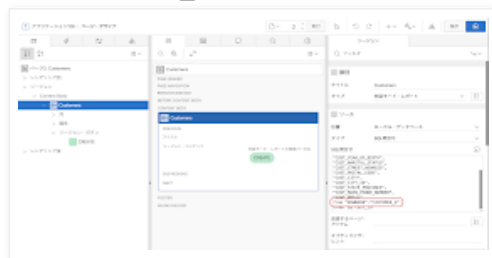
インポート済みのCustomer Service AppをOracle APEXのアプリケーション・ビルダーから開き、編集作業を開始します。



編集対象の対話レポートを含む、ページ2のCustomersを開きます。



対話レポートで使用されているビューがCUSTOMER_Vであることを確認します。



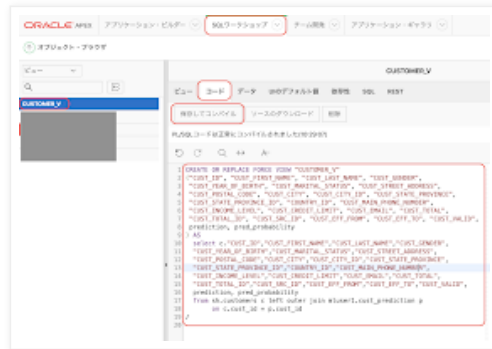
ビューの定義を以下に置き換えます。顧客情報に機械学習を使用して予測した結果であるPREDICTION列とPRED_PROBABILITY列を追加します。SELECT文にある"CUST_ID"をc."CUST_ID"に変更している点に注意してください。

```
CREATE OR REPLACE FORCE VIEW "CUSTOMER_V"
("CUST_ID", "CUST_FIRST_NAME", "CUST_LAST_NAME", "CUST_GENDER",
"CUST_YEAR_OF_BIRTH", "CUST_MARITAL_STATUS", "CUST_STREET_ADDRESS",
"CUST_POSTAL_CODE", "CUST_CITY", "CUST_CITY_ID", "CUST_STATE_PROVINCE",
"CUST_STATE_PROVINCE_ID", "COUNTRY_ID", "CUST_MAIN_PHONE_NUMBER",
```

```

"CUST_INCOME_LEVEL", "CUST_CREDIT_LIMIT", "CUST_EMAIL", "CUST_TOTAL",
"CUST_TOTAL_ID", "CUST_SRC_ID", "CUST_EFF_FROM", "CUST_EFF_TO", "CUST_VALID",
prediction, pred_probability
) AS
select c."CUST_ID","CUST_FIRST_NAME","CUST_LAST_NAME","CUST_GENDER",
"CUST_YEAR_OF_BIRTH","CUST_MARITAL_STATUS","CUST_STREET_ADDRESS",
"CUST_POSTAL_CODE" "CUST_CITY","CUST_CITY_ID","CUST_STATE_PROVINCE",
"CUST_STATE_PROVINCE_ID","COUNTRY_ID","CUST_MAIN_PHONE_NUMBER",
"CUST_INCOME_LEVEL", "CUST_CREDIT_LIMIT", "CUST_EMAIL", "CUST_TOTAL",
"CUST_TOTAL_ID", "CUST_SRC_ID", "CUST_EFF_FROM", "CUST_EFF_TO", "CUST_VALID",
prediction, pred_probability
from sh.customers c left outer join mluser1.cust_prediction p
on c.cust_id = p.cust_id
/

```

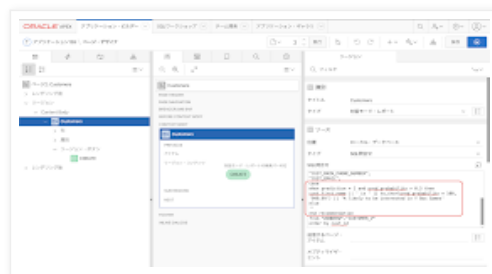


APEXアプリケーションに戻り、ページ2の対話レポートのSQLを編集します。SELECT文に以下の列RECOMMENDATIONを追加します。"CUST_EMAIL"の後ろにカンマを追加するのを忘れないようにしましょう。

```

case
when prediction = 1 and pred_probability > 0.5 then
cust_first_name || ' is ' || to_char(pred_probability * 100, '999.99') || '% likely to be interested in Y Box Games'
else
''
end recommendation

```



機械学習による予測を追加したAPEXアプリケーションを確認する

更新したアプリケーションを実行し、顧客リストの対話レポートを表示します。Recommendation列に何も表示されていない行が多いので、Y Box Gamesの販売対象となっている顧客はあまり存在していません。これらの顧客は、この製品に興味はもたないでしょう。

名(Cust First Name)が"Connor"、姓(Cust Last Name)が"Clark"の顧客を検索します。1970年生まれのConnor Clarkさんの行にはRecommendation列の表示があり、Y Box Gamesを高い確率で気に入るだろうと案内されています。

[詳細プロフィールを表示](#)