Practice: Using Sample Data to Explore Visualizations, Predictions, and Classification Algorithms (Use Case)

Get your free cloud account: click here.

Overview

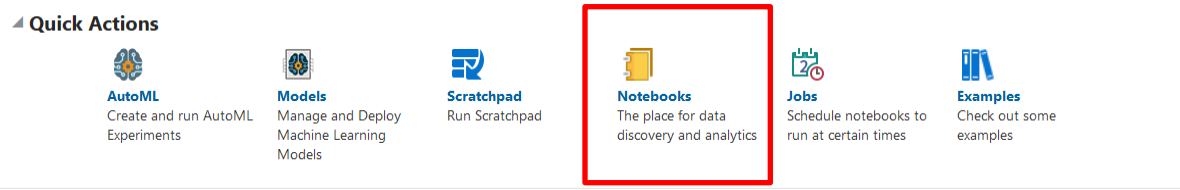
In this practice, you use sample data to explore visualizations using Classification Prediction.

Use Case: Predicting Target Customers Using Classification

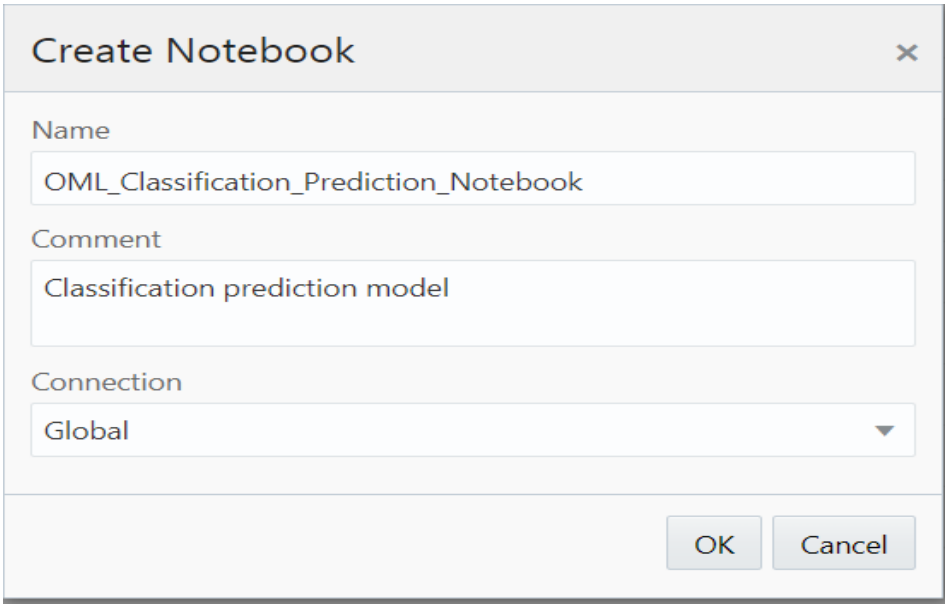
Create a notebook to predict customers most likely to be positive responders to an Affinity Card loyalty program. This notebook builds and applies classification models (decision tree) using the SH schema data and is processed inside Oracle Autonomous Data Warehouse (ADW).

Tasks

1. Log into your Oracle Cloud Free Tier Account
2. On the Oracle Machine Learning home page, click **Notebooks**.



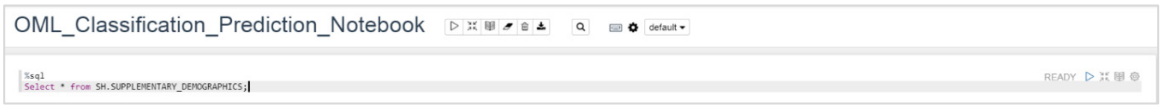
3. In the **Notebooks** action item, click **Create**.
4. In the **Create Notebook** window, enter **Name**, **Comment,** and **Global** as a connection string. After you click the **OK** button, you will get your notebook ready.



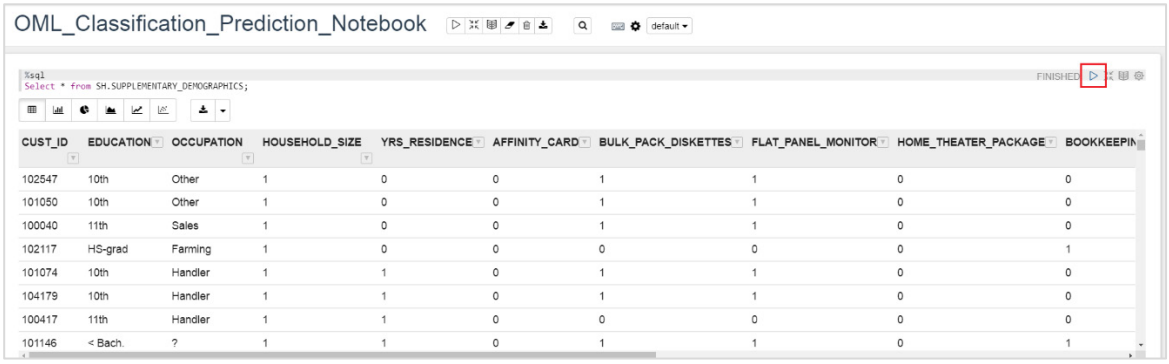
5. In the **OML_Classification_Prediction_Notebook** area, copy and paste the following code:

```
%sql
Select * from SH.SUPPLEMENTARY_DEMOGRAPHICS;
```

   a. Your screen should now look like this:



   b. Click the **Run this paragraph** icon shown below to execute the SQL statement and display the results in a tabular format.



CUST_ID	EDUCATION	OCCUPATION	HOUSEHOLD_SIZE	YRS_RESIDENCE	AFFINITY_CARD	BULK_PACK_DISKETTES	FLAT_PANEL_MONITOR	HOME_THEATER_PACKAGE	BOOKKEEPIN
102547	10th	Other	1	0	0	1	1	0	0
101050	10th	Other	1	0	0	1	1	0	0
100040	11th	Sales	1	0	0	1	1	0	0
102117	HS-grad	Farming	1	0	0	0	0	0	1
101074	10th	Handler	1	1	0	1	1	0	0
104179	10th	Handler	1	1	0	1	1	0	0
100417	11th	Handler	1	1	0	0	0	0	0
101146	< Bach.	?	1	1	0	1	1	0	1

6. Changing the report type

   a. Using the **report menu bar**, you can change the table to a graph and/or export the result set to a CSV or TSV file.

   b. Click the **bar graph** icon and then the **Settings** link to unfold the settings panel for the graph.

      Note: To add a column to one of the **Keys**, **Groups,** or **Values** panels, just drag and drop the column name into the required panel. To remove a column from the Keys, Groups, or Values panel, just click the **x** next to the column name displayed in the relevant panel.

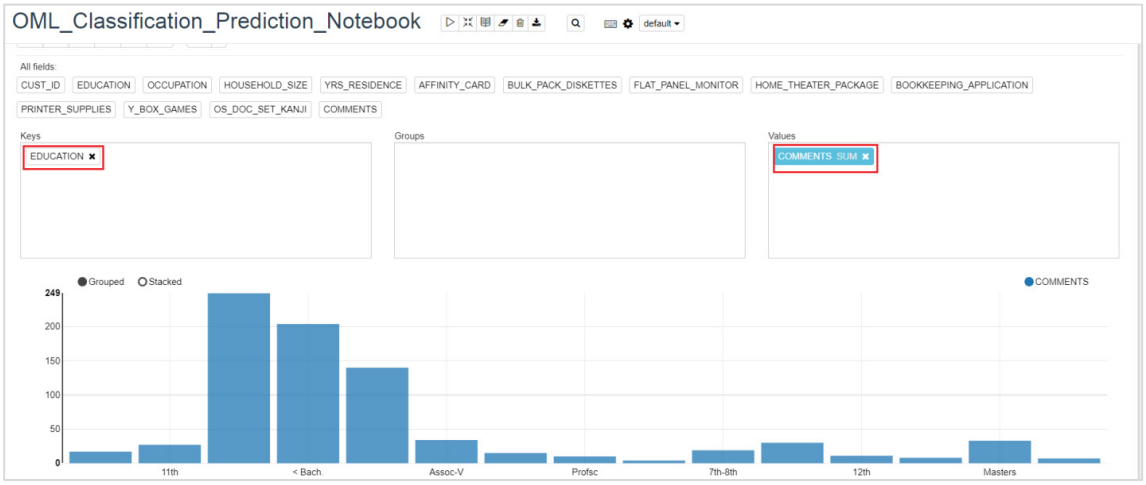
7. Changing the layout of the graph

   a. With the graph settings panel visible, remove all the columns from both the Keys and Values panels.

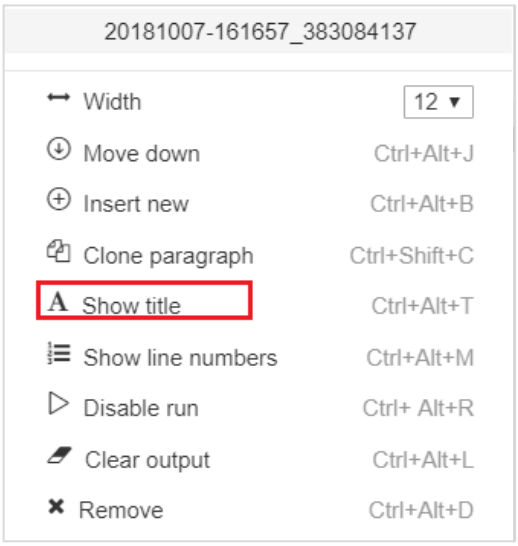
   b. Drag and drop EDUCATION into the Keys panel.

   c. Drag and drop COMMENTS SUM into the Values panel.

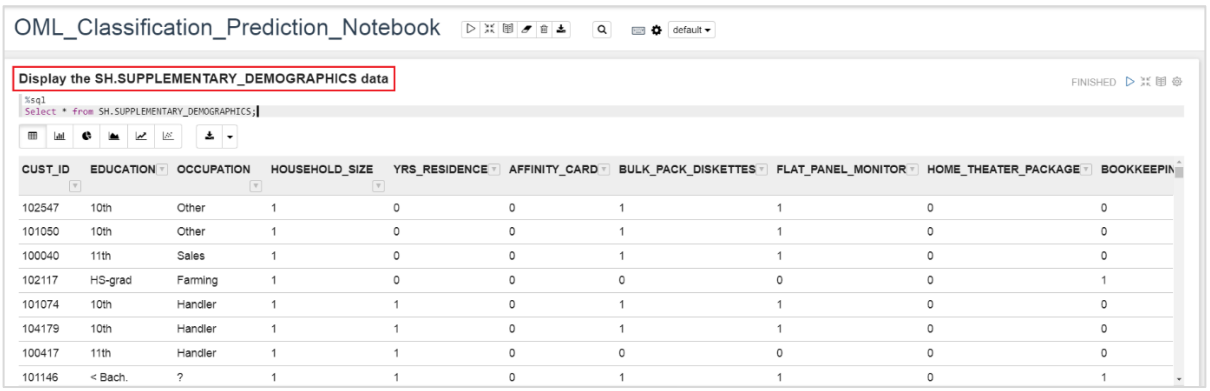
d. The report should now look like the one shown below.



8. To show the title for the current paragraph, click ⚙ in the right corner and select **Show title**.

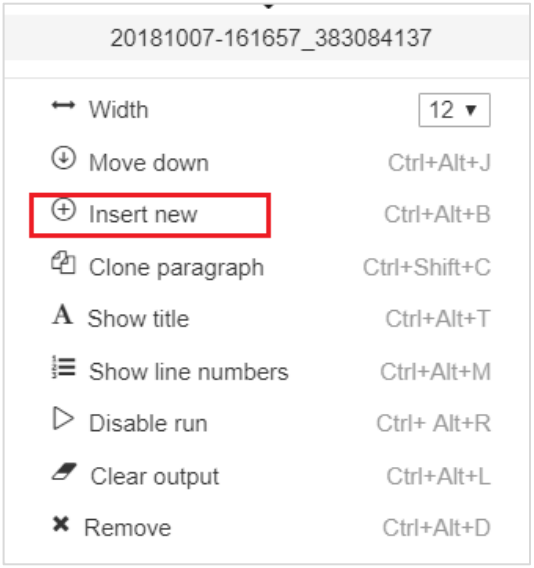


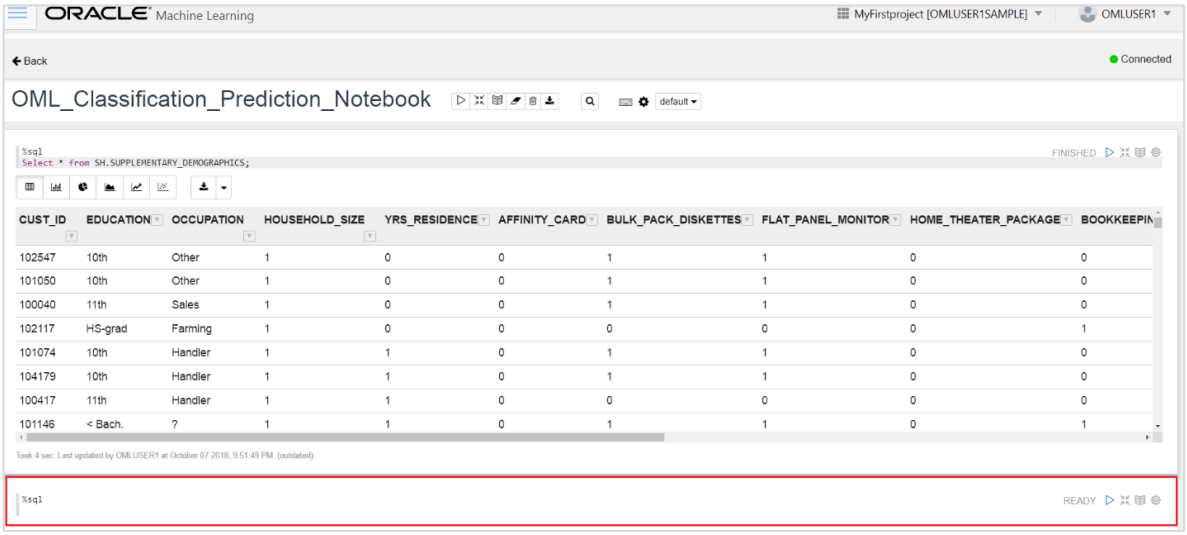
9. Name the title (Display the SH.SUPPLEMENTARY_DEMOGRAPHICS Data) in the paragraph, as shown below. The data is displayed in the Table option.



CUST_ID	EDUCATION	OCCUPATION	HOUSEHOLD_SIZE	YRS_RESIDENCE	AFFINITY_CARD	BULK_PACK_DISKETTES	FLAT_PANEL_MONITOR	HOME_THEATER_PACKAGE	BOOKKEEPIN
102547	10th	Other	1	0	0	1	1	0	0
101050	10th	Other	1	0	0	1	1	0	0
100040	11th	Sales	1	0	0	1	1	0	0
102117	HS-grad	Farming	1	0	0	0	0	0	1
101074	10th	Handler	1	1	0	1	1	0	0
104179	10th	Handler	1	1	0	1	1	0	0
100417	11th	Handler	1	1	0	0	0	0	0
101146	< Bach.	?	1	1	0	1	1	0	1

10. To add one more paragraph in the notebook, click  and select **Insert new**. A new paragraph will be created.





11. Title the paragraph “Graph HOUSEHOLD_SIZE from SH.SUPPLEMENTARY_DEMOGRAPHICS Data” with the help of the previous steps, and copy and paste the following query in the second paragraph:

```
%sql
Select count(cust_id) as TOTAL, household_size, affinity_card
from SH.SUPPLEMENTARY_DEMOGRAPHICS group by household_size,
affinity_card;
```

a. Click the **Run this paragraph** icon shown below to execute the SQL statement and display the execution message.

Because the title of this paragraph is Graph Household …, display the data by using the visualize option. Click Bar Chart and you will see the following:

b. Create the paragraphs (in **OML_Classification_Prediction_Notebook**) one by one with the tile name, copy and paste the following SQL statements, and run them one by one to see the execution messages.

Paragraph 3:

Tile Name: Clean Up and Drop the Table If It Already Exists and Create a Data Table

```
%script
-- Clean out old table
BEGIN
      EXECUTE IMMEDIATE 'DROP Table
SUPPLEMENTARY_DEMOGRAPHICS2';
  EXCEPTION
      WHEN OTHERS THEN NULL;
END;
/
```

Create a SUPPLEMENTARY_DEMOGRAPHICS2 table

```
CREATE Table SUPPLEMENTARY_DEMOGRAPHICS2
  AS (SELECT AFFINITY_CARD,
BOOKKEEPING_APPLICATION, BULK_PACK_DISKETTES,
CUST_ID, EDUCATION, FLAT_PANEL_MONITOR,
HOME_THEATER_PACKAGE, HOUSEHOLD_SIZE, OCCUPATION,
OS_DOC_SET_KANJI, PRINTER_SUPPLIES, YRS_RESIDENCE,
Y_BOX_GAMES
```

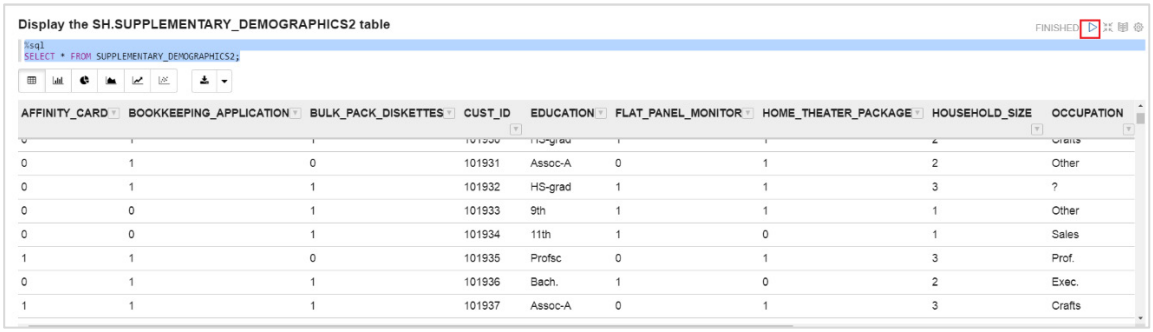
Paragraph 4:

Tile Name: Display the SH.SUPPLEMENTARY_DEMOGRAPHICS2 Table

Query:

```
%sql

SELECT * FROM SUPPLEMENTARY_DEMOGRAPHICS2;
```



AFFINITY_CARD	BOOKKEEPING_APPLICATION	BULK_PACK_DISKETTES	CUST_ID	EDUCATION	FLAT_PANEL_MONITOR	HOME_THEATER_PACKAGE	HOUSEHOLD_SIZE	OCCUPATION
0	1	0	101931	Assoc-A	0	1	2	Other
0	1	1	101932	HS-grad	1	1	3	?
0	0	1	101933	9th	1	1	1	Other
0	0	1	101934	11th	1	0	1	Sales
1	1	0	101935	Profsc	0	1	3	Prof.
0	1	1	101936	Bach.	1	0	2	Exec.
1	1	1	101937	Assoc-A	0	1	3	Crafts

Paragraph 5:

Tile Name: Preparatory Steps, Automation of Model Build, and Test and Clean-Up Using PL/SQL Script

Query:

```
%script
--Build a classification model and then generate a lift test
result and an apply result.
DECLARE
v_sql varchar2(100);
BEGIN
-- drop build settings
BEGIN
v_sql := 'DROP TABLE n1_build_settings PURGE';
EXECUTE IMMEDIATE v_sql;
DBMS_OUTPUT.PUT_LINE (v_sql ||': succeeded');
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE (v_sql ||': drop unnecessary - no table
exists');
END;
```

Drop Model

```
BEGIN
v_sql := 'CALL DBMS_DATA_MINING.DROP_MODEL(''N1_CLASS_MODEL'')';
EXECUTE IMMEDIATE v_sql;
DBMS_OUTPUT.PUT_LINE (v_sql ||': succeeded');
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE (v_sql ||': drop unnecessary - no model
exists');
END;
```

drop apply result

```
BEGIN
v_sql := 'DROP TABLE N1_APPLY_RESULT PURGE';
EXECUTE IMMEDIATE v_sql;
DBMS_OUTPUT.PUT_LINE (v_sql ||': succeeded');
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE (v_sql ||': drop unnecessary - no table
exists');
END;
```

drop lift result

```
BEGIN
v_sql := 'DROP TABLE N1_LIFT_TABLE PURGE';
EXECUTE IMMEDIATE v_sql;
DBMS_OUTPUT.PUT_LINE (v_sql ||': succeeded');
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE (v_sql ||': drop unnecessary - no table
exists');
END;
```

Split the Data into N1_TRAIN_DATA and N1_TEST_DATA

```
EXECUTE IMMEDIATE 'CREATE OR REPLACE VIEW N1_TRAIN_DATA AS
SELECT * FROM SUPPLEMENTARY_DEMOGRAPHICS2 SAMPLE (60) SEED
(1)';
DBMS_OUTPUT.PUT_LINE ('Created N1_TRAIN_DATA');
EXECUTE IMMEDIATE 'CREATE OR REPLACE VIEW N1_TEST_DATA AS
SELECT * FROM SUPPLEMENTARY_DEMOGRAPHICS2 MINUS SELECT * FROM
N1_TRAIN_DATA';
DBMS_OUTPUT.PUT_LINE ('Created N1_TEST_DATA');
```

Create a Build Setting (DT) for Model Build

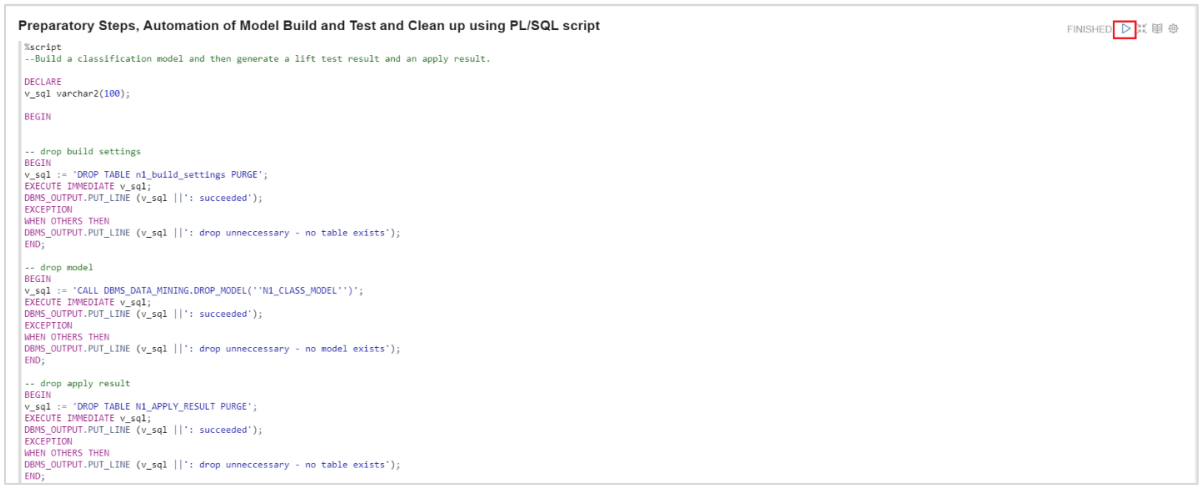
```
EXECUTE IMMEDIATE 'CREATE TABLE n1_build_settings (setting_name
VARCHAR2(30),setting_value VARCHAR2(4000))';
EXECUTE IMMEDIATE 'INSERT INTO n1_build_settings (setting_name,
setting_value) VALUES (''ALGO_NAME'', ''ALGO_DECISION_TREE'')';
EXECUTE IMMEDIATE 'INSERT INTO n1_build_settings (setting_name,
setting_value) VALUES (''PREP_AUTO'', ''ON'')';
DBMS_OUTPUT.PUT_LINE ('Created model build settings table:
n1_build_settings ');
```

Build a Classification Model

```
EXECUTE IMMEDIATE 'CALL
  DBMS_DATA_MINING.CREATE_MODEL(''N1_CLASS_MODEL'',
  ''CLASSIFICATION'', ''N1_TRAIN_DATA'', ''CUST_ID'',''
  AFFINITY_CARD'', ''n1_build_settings'')';
DBMS_OUTPUT.PUT_LINE ('Created model: N1_CLASS_MODEL ');
```

Test the Model by Generating an Apply Result and Then Create a Lift Result

```
EXECUTE IMMEDIATE 'CALL
DBMS_DATA_MINING.APPLY(''N1_CLASS_MODEL'',''N1_TEST_DATA'',''CUST
_ID'',''N1_APPLY_RESULT'')';
DBMS_OUTPUT.PUT_LINE ('Created apply result: N1_APPLY_RESULT ');
EXECUTE IMMEDIATE 'CALL
DBMS_DATA_MINING.COMPUTE_LIFT(''N1_APPLY_RESULT'',''N1_TEST_DATA'
',''CUST_ID'',''AFFINITY_CARD'',''N1_LIFT_TABLE'',''1'',''PREDICT
ION'',''PROBABILITY'',100)';
DBMS_OUTPUT.PUT_LINE ('Created lift result: N1_LIFT_TABLE ');
END;
```

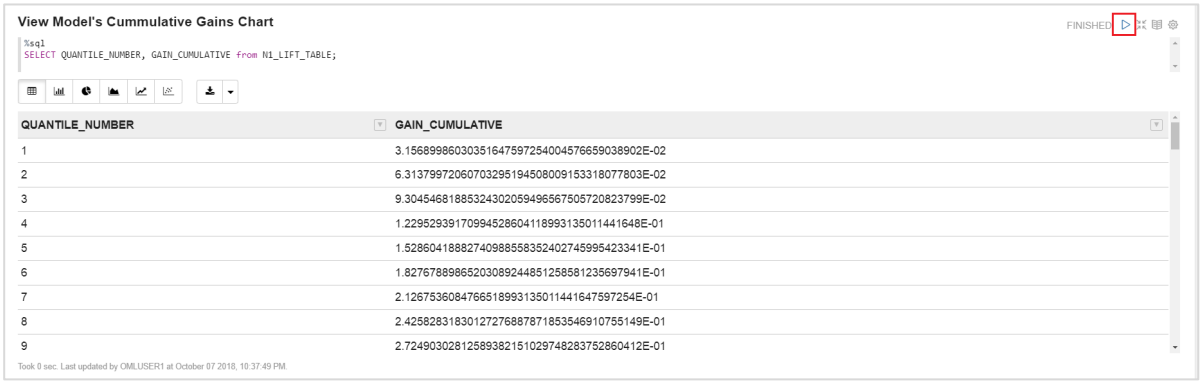


Paragraph 6:

Tile Name: View the Model's Cumulative Gains Chart

Query:

```
%sql
SELECT QUANTILE_NUMBER, GAIN_CUMULATIVE from N1_LIFT_TABLE;
```



View Model's Cummulative Gains Chart

FINISHED

```
%sql
SELECT QUANTILE_NUMBER, GAIN_CUMULATIVE from N1_LIFT_TABLE;
```

QUANTILE_NUMBER	GAIN_CUMULATIVE
1	3.156899860303516475972540045766590389O2E-02
2	6.313799720607032951945080091533180778O3E-02
3	9.304546818853243020594965675057208237 99E-02
4	1.229529391709945286041189931350114416 48E-01
5	1.528604188827409885583524027459954233 41E-01
6	1.827678898652030892448512585812356979 41E-01
7	2.126753608476651899313501144164759725 4E-01
8	2.425828318301272768878718535469107551 49E-01
9	2.724903028125893821510297482837528604 12E-01

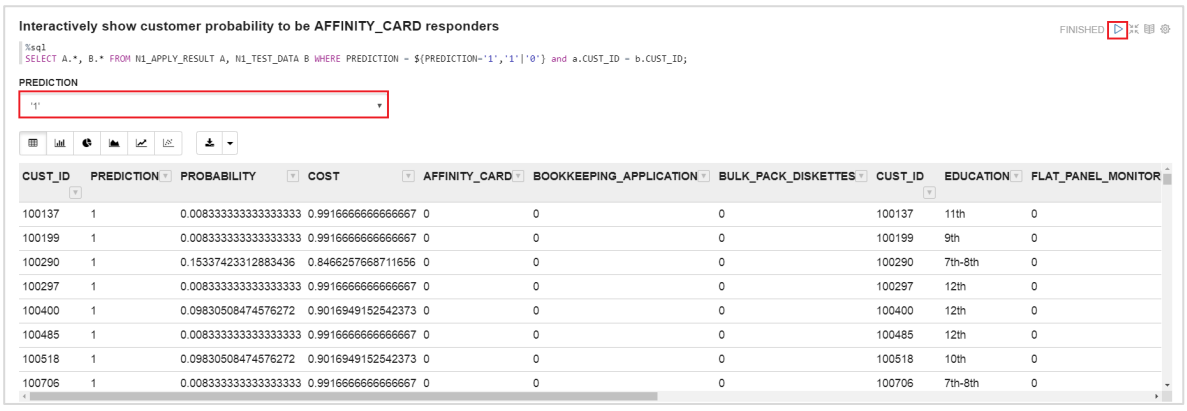
Took 0 sec. Last updated by OMLUSER1 at October 07 2018, 10:37:49 PM.

Paragraph 7:

Tile Name: Interactively Show Customer Probability to be AFFINITY_CARD Responders

Query:

```
%sql
SELECT A.*, B.* FROM N1_APPLY_RESULT A, N1_TEST_DATA B WHERE
PREDICTION = ${PREDICTION='1','1'|'0'} and a.CUST_ID = b.CUST_ID;
```



Interactively show customer probability to be AFFINITY_CARD responders

FINISHED

```
%sql
SELECT A.*, B.* FROM N1_APPLY_RESULT A, N1_TEST_DATA B WHERE PREDICTION = ${PREDICTION='1','1'|'0'} and a.CUST_ID = b.CUST_ID;
```

PREDICTION

'1'

CUST_ID	PREDICTION	PROBABILITY	COST	AFFINITY_CARD	BOOKKEEPING_APPLICATION	BULK_PACK_DISKETTES	CUST_ID	EDUCATION	FLAT_PANEL_MONITOR
100137	1	0.008333333333333333	0.9916666666666667	0	0	0	100137	11th	0
100199	1	0.008333333333333333	0.9916666666666667	0	0	0	100199	9th	0
100290	1	0.15337423312883436	0.8466257668711656	0	0	0	100290	7th-8th	0
100297	1	0.008333333333333333	0.9916666666666667	0	0	0	100297	12th	0
100400	1	0.09830508474576272	0.9016949152542373	0	0	0	100400	12th	0
100485	1	0.008333333333333333	0.9916666666666667	0	0	0	100485	12th	0
100518	1	0.09830508474576272	0.9016949152542373	0	0	0	100518	10th	0
100706	1	0.008333333333333333	0.9916666666666667	0	0	0	100706	7th-8th	0

Paragraph 8:

Tile Name: Interactive Pie Chart of Likely AFFINITY_CARD Responders

Query:

```
%sql
SELECT A.*, B.* FROM N1_APPLY_RESULT A, N1_TEST_DATA B WHERE
PREDICTION = ${PREDICTION='1','1'|'0'} and a.CUST_ID = b.CUST_ID;
```

Note: Remember to click the Pie Chart option to see the data in the above format.

Paragraph 9:

Tile Name: Select Customers Who Are Above the Threshold Probability (20%) of Being Likely AFFINITY_CARD Responders

Query:

```
%sql

SELECT * from(
SELECT CUST_ID, PREDICTION_PROBABILITY(N1_CLASS_MODEL, '1'
USING A.*) prob
 FROM N1_APPLY_RESULT A)
 WHERE prob > 0.2;
```

CUST_ID	PROB
102800	0.23255813953488372
102900	0.23255813953488372
103600	0.23255813953488372
100011	0.23255813953488372
100050	0.23255813953488372
100097	0.23255813953488372
100264	0.23255813953488372
100321	0.23255813953488372
100617	0.23255813953488372

Paragraph 10:

Tile Name: Real-Time Prediction

Query:

```
%sql
Select prediction_probability(N1_CLASS_MODEL, '1'
  USING '3' as HOUSEHOLD_SIZE, 5 as YRS_RESIDENCE, 1 as
Y_BOX_GAMES)
from dual;
```

Paragraph 11:

Tile Name: Interactive Selection of Likely Affinity_Card Responders Selected by HOUSEHOLD_SIZE

Query:

```
%sql
SELECT A.*, B.* FROM N1_APPLY_RESULT A, N1_TEST_DATA B WHERE
HOUSEHOLD_SIZE = ${HOUSEHOLD_SIZE='1','1'|'3'| '9+'} and
a.CUST_ID = b.CUST_ID;
```

CUST_ID	PREDICTION	PROBABILITY	COST	AFFINITY_CARD	BOOKKEEPING_APPLICATION	BULK_PACK_DISKETTES	CUST_ID	EDUCATION	FLAT_PANEL_MONITC
100551	0	0.9857142857142858	0.014285714285714235	0	1	0	100551	HS-grad	0
100551	1	0.014285714285714285	0.9857142857142858	0	1	0	100551	HS-grad	0
100619	0	0.9857142857142858	0.014285714285714235	0	1	0	100619	HS-grad	0
100619	1	0.014285714285714285	0.9857142857142858	0	1	0	100619	HS-grad	0
100631	0	0.916095890410959	0.08390410958904104	0	1	0	100631	< Bach.	0
100631	1	0.0839041095890411	0.9160958904109588	0	1	0	100631	< Bach.	0
100640	0	0.9857142857142858	0.014285714285714235	0	1	0	100640	HS-grad	0
100640	1	0.014285714285714285	0.9857142857142858	0	1	0	100640	HS-grad	0

You have successfully created a notebook to predict customers most likely to be positive responders to an Affinity Card loyalty program.

This completes the practice for using sample data to explore visualizations using Classification Prediction.