Practice: Exploring the Bank Promotion Data Set Using CAS and the R API

A large financial institution created a new product for its customers. They marketed the product to random customers and gathered demographic and financial information from these customers. Your goal is to build a model to predict which customers are most likely to purchase the new product.

The **BANK_PROMO** data set contains information from account holders at a large financial services firm. The accounts represent consumers of home equity lines of credit, automobile loans, and other short-to-medium-term credit instruments.

Name	Model Role	Measurement Level	Description
B_TGT	Target	Binary	1 = customer purchased new product, 0 = customer did not purchase
CAT_INPUT1	Input	Nominal	Account activity level
CAT_INPUT2	Input	Nominal	Customer Value level
DEMOG_AGE	Input	Interval	Customer age
DEMOG_GEN	Input	Binary	Customer gender
DEMOG_HOS	Input	Binary	Homeowner status
DEMOG_HOMEVAL	Input	Interval	Home value
DEMOG_INC	Input	Interval	Income
RFM5	Input	Interval	Purchase count past three years
RFM6	Input	Interval	Purchase count lifetime
RFM7	Input	Interval	Purchase count past three years direct promotion response
RFM8	Input	Interval	Purchase count lifetime direct promotion response
RFM9	Input	Interval	Months since last purchase
RFM10	Input	Interval	Total count promos past year
RFM11	Input	Interval	Direct promos count past year
RFM12	Input	Interval	Customer Tenure

1. From the Jupyter Lab Home directory, select the plus symbol to open a new page and then select R under Notebook to create a new notebook.

2. Load the SWAT, GGPLOT2, RESHAPE2, and REPR packages. Use the options function to change the height and width of graphics to 5.5.

3. Using the CAS function, connect to CAS and create a connection object called **conn**.

4. Use the time-out action from the session action set to change the time-out to 12 hours (60*60*12=43200 seconds).

5. Use the cas.read.csv function to load **bank_promo.csv** onto the server and create a CAS table object reference for the data. Name the new data set **bank** and create a variable to reference the data table in subsequent code.

6. Use SWAT functionality to explore the data. Use the head, dim, names, colMeans, and summary functions on the CAS table object.

7. In the summary function output, notice that the minimum age is 6. In the **bank_promo** data set, any age below 18 is an error. Load the dataStep action set and run the following code to replace all nonmissing **demog_age** values less than 18 with 18 years of age. Then run the summary function on the **demog_age** variable to ensure that only the variable was transformed.

```
cas.dataStep.runCode(conn, code=
    "
    data bank;
        set bank;
        if demog_age < 18 and demog_age ^= . then demog_age = 18;
    run;
    "
)
```

8. Use CAS actions instead of SWAT functionality to explore the data. Load the simple action set and then use the correlation action to find the correlations among the eight different **rfm** variables.

9. Use the distinct action from the simple action set to find the number of levels for the variables **b_tgt**, **cat_input1**, **cat_input2**, **demog_gen**, and **demog_hos**.

10. Use the freq action from the simple action set to find the frequency of each level and the number of missing values for the variables **b_tgt**, **cat_input1**, **cat_input2**, **demog_gen**, and **demog_hos**.

11. Use the crossTab action from the simple action set to find a cross tabulation of the variables **b_tgt** and **cat_input2**.

12. Load the cardinality action set and use the summarize action on the **bank** data table. Save the results to a new data table called **card** on the server.

13. Create an object reference for the **card** data table and then use the head and nrow SWAT functions to view the first few rows and the number of rows of the new data table.

14. Sample 25% of the **bank** data by first loading the sampling action set and then using the srs action. Name the new sample data table **mysam** on the server.

15. Use the defCasTable function to create an object reference to the **mysam** table. Then use the to.casDataFrame function to download the in-memory table to the client. Finally, use the melt function from the reshape2 package to format the data table, and using ggplot, create a panel of histograms for the variables.

16. Use the distinct action from the simple action set to create a table of the variable name and number of missing observations for each. Setting the action equal to an object creates a copy of the table on the client. Next, create a column of the percentage of missing values for each variable and plot the percentages with ggplot.

17. All the inputs in the **bank** data set have missing values. Use the impute action from the dataPreprocess action set to impute the continuous inputs with the median value and the nominal inputs with the mode.

18. Create variable shortcuts. Either create the target, inputs, and nominals manually, or subset the variable names by the variable type and the IMP_ prefix by first using the columnInfo action from the table action set.