Campaign Management Template with R Scripts

This is the R (Microsoft R Server) code for Campaign Management template using SQL Server R Services. This code runs on a local R IDE (such as RStudio, R Tools for Visual Studio), and the computation is done in SQL Server (by setting compute context).

This is primarily for customers who prefer advanced analytical solutions on a local R IDE

It consists of the following files:

|  |  |
| --- | --- |
| File | Description |
| Step1\_input\_data.R | Simulates the 4 input datasets |
| Step2\_data\_preprocessing.R | Performs preprocessing steps like outlier treatment and missing value treatment on the input datasets |
| Step3\_feature\_engineering\_AD\_creation.R | Performs Feature Engineering and creates the Analytical Dataset |
| Step4\_model\_rf\_gbm.R | Builds the Random Forest & Gradient Boosting models, identifies the champion model and scores the Analytical dataset |

Note: The connection parameters are not set in any of the scripts. The user will have to enter these parameters in the beginning of each script before running them.

**Step1\_input\_data.R**

This script simulates the 4 input datasets and exports to SQL Server. The user also needs to input the number of leads that need to be simulated by entering the value in line 29 of the script

1. Lead Demography: Based on the number of leads entered in line 29, the script creates Hexadecimal Lead Ids and simulates other variables like age, annual income, credit score, location, educational background and many other demographic details for each lead. The script generates some missing values so they can be treated later in pre-processing
2. Market Touchdown: Every lead’s lead Id, age, annual income & credit scores are extracted from the Lead Demography table and variables from historical campaign data is simulated here by applying randomization. A few outliers are created here intentionally, so that they can later be handled in pre-processing
3. Campaign Detail: In this part of the script, the Campaign metadata like campaign name, launch date, category, sub-category are simulated
4. Product: In this part of the script, the Product metadata like product name, category, term, premium etc are simulated

**Step2\_data\_preprocessing.R**

This script performs missing value and outlier treatment on the lead demography and market touchdown tables. Both these updated tables are then exported back to SQL Server

1. Market Touchdown: The Communication latency variable in this table was created to have outliers. The lower extremes are replaced with the difference of Mean and Standard Deviation. The higher extremes are replaced with the sum of Mean and two Standard Deviations
2. Lead Demography: The missing values in variables like number of children/dependents, highest education & household size are replaced with the Mode value

**Step3\_feature\_engineering\_AD\_creation.R**

This scripts performs feature engineering on the Market Touchdown table and then merges the 4 input tables to generate the Analytical Dataset. Finally, the analytical dataset along with training and test datasets are exported to SQL Server

1. Market Touchdown: The table is aggregated at a lead level, so variables like channel which will have more than one value for each user are pivoted and aggregated to from variables like SMS count, Email count, Call Count, Last Communication Channel, Second Last Communication Channel etc.
2. Analytical Dataset: The latest version of all the 4 input datasets are merged together to create the analytical dataset. The analytical dataset is further split into train and test datasets

**Step4\_model\_rf\_gbm.R**

In this step, two models are built using 2 statistical techniques on the training Dataset. Once the models are trained, AUC of both the models are calculated using the test dataset. The model with the best AUC is selected as the champion model