Optimization of the promotion expense using Market mix modeling–Multi product Insurance Industry Case:

Case Synopsis: Mr. A is a marketing head of a Multi product Insurance Company. He manages the funds allocation for various promotional activities. He has a large database of general insurance (GI) , Life and health insurance customers. He wants to calculate the ROI of his promotional activities and optimize it by thoughtful allocation across the different products. He has the historical data for products performance and ROI for campaigns/promotions. He managed and monitored the campaigns performance using test and control groups in the past. He can use this data now for developing the advanced models.

He needs to build the market mix model to identify the contributing factors for high performing brands and estimate the ROI of the campaigns.

The implementation of the same will involve design and development of a market mix model.

Background:

Market mix modeling is a widely used marketing model to evaluate the return on investment (ROI) of various marketing components like price, promotion, product quality etc. Statistically it's a form of data mining, the "analysis of observational datasets to find unforeseen relationships and to summarize the data in ways that are useful to the data owner."

Market mix modeling finds the relationship among various independent variables and the dependent variable and use them to make inferences about the required composition in future. The market mix models can use different types of modeling relations for eg. Additive, multiplicative and other transformations based on the available data and business scenario. The appropriate model depends on the business scenario and the data.

Market mix modeling combines the forecasting and regression techniques to deliver the results.

The multiplicative model is log transformed for regression.
The model takes the following form:

$$Yt = \alpha + \beta 1\, At + \beta 2 Pt + \beta 3 Rt + \beta 4 Qt + \varepsilon t.$$

Here, Y represents the dependent variable (e.g.,sales), while the other capital letters represent variables of the marketing mix, such as advertising , (A), price (P), sales promotion (R), or quality (Q). The parameters α and βk are coefficients that the researcher wants to estimate. βk represents the effect of the independent variables on the dependent variable, where the subscript k is an index for the independent variables. The subscript t represents various time periods.
The multiplicative model derives its name from the fact that the independent variables of the marketing mix are multiplied together.

$$Yt = Exp(\alpha) \times At\beta 1 \times Pt\beta 2 \times Rt\beta 3 \times Qt\beta 4 \times \varepsilon t.$$

$\log (Yt) = \alpha + \beta 1 \log(At) + \beta 2 \log(Pt) + \beta 3 \log(Rt) + \beta 4 \log(Qt) + \varepsilon t.$

Modeling requires data from various sources and the input data extracted is very important to build a robust model. We need to collect the sales data, promotion data, product data and the historical information in the data and do some initial analysis like data audit, exploratory data analysis and the transformations for better understanding of the data. Next step involves analysis of the relationship in the data. The key objective of analysis is to establish a relation between the dependent and independent variables using regression. Once the model is built on the treated data we need to perform the calculations to use the model.

Workflow

Pull the relevant variables from the database. This step is called as Data Extraction.

The data is extracted by writing the SQL codes. The same is imported to the R for further analysis.

The initial dataset used is extracted after manipulations of marketing data and comprises of sales data, promotion indicator and pricing details. The figures are log transformed for modeling purpose.

The details of the data preparation is covered in the data preparation part.

See the excel/csv file for data.

Initial Data Analysis:

Before Data modeling, we should go through the data for better understanding of the available data and the business problem. Meantime it will helps in identifying missing values and outliers if there in your data.

Code

```
##Set working directory
setwd("E:/documents/SelfStudy/market mix model/Final deliverable/R MMM")

##Read data
data <-read.csv("MMM_ds_1.csv", stringsAsFactors = TRUE, strip.white = TRUE, na.strings =
c("NA",""))

##inspect the imported data
str(data)
head(data)
tail(data)

##Generate Summary of Data
summary(data)/*Verification for missing value treatment*/

## impute missing values
```

```
is.na(data) #Checking Missing Values exist or not

##When data volume is huge list rows of data that have missing values
data[!complete.cases(data),]

# Recode Missing to 1
data$cmpgn1[is.na(data$cmpgn1)] <- 1
data$campgn2[is.na(data$campgn2)] <- 1
```

The data preparation steps involve following steps:

1. Data Audit and EDA
2. Data Profiling

There are certain criteria that we should ensure before considering Market mix modeling:

1) Dependency of the sales on the product and promotion data
2) Price of the product should impact sales.
3) Sales data for both test and control group should be available. The sales data for the control group is used as baseline sales. The seasonality factor is used to capture seasonal effects on the sales.

Data Audit and EDA: - The data audit report is the initial report that we prepared to understand the data well. This report will consists of descriptive statistics for all the variables in the dataset and also will helps in indentifying the missing value and levels of categorical predictive variables. This data Audit report serves as base for assessing the quality of the data we extracted and obtained from the client. Based on this report we can request for additional data which we seems to be important for our analysis and it will also helps in dropping some insignificant variable. Refer the data audit report fields below for better understanding.

Variable
Data Type
Label
Total no. of records in dataset
No. of valid (non missing) cases
No. of missing values
Fill Rate
Count of Unique Values
No. of Levels
Mean
Std deviation
Median
Mode
Minimum
Q1(25th percentile)
Q3(75th Percentile)
99th Percentile
Maximum

Data Profiling:-Bivariate profiling assist in finding the frequency of each categorical variable with respect to the response variable. This would facilitate in binding/ grouping the categories which have same response rate so that the effect of that particular category can be captured in the model.

We prepare a report with following fields to further work on it.

Variable
Field categories
Frequency
Responses
Responses %

Data Treatment:

The first stage of any statistical modeling consists of data treatment activities. Approximately 80% of the entire modeling time is consumed by the data treatment techniques. Here we check the hygiene factor of our independent variables and try to make the data as exploitable as possible.

Before going to data treatment one has to find out the correlation between variables means finding the relationship of predictors with the response variable and also to find out the inter correlation among predictors. From this analysis we can exclude some of the predictors which

are not important for the model building based on the significant correlation values. The first step of variable reduction happens in this stage and next is on basis of multicollinearity check. The variables selected from this step will undergo for further data treatment like missing value, extreme value treatment and multicollinearity check.

Missing value treatment: -

We should check that the independent variables have sufficient information to establish a significant relation with the dependent variables.

Some of the basic rules for Missing value treatment are as below:

1. If the independent variables have a large amount of missing value (More than 40%-50%), we drop that independent variable from our analysis, since no relation can be established between that independent variable and the dependent variable in question.
2. If the percentage of missing value lies between 10% -40%, we try to establish a separate relation between the dependent and independent variables to understand any hidden pattern.
3. If our predictors are categorical variable, then we can make that missing values as one category but we will miss the information since that category will not comes significant in the model so better treat the missing value with those category which has highest frequency among all the categories of a variable.
4. For quantitative independent variable, treat the missing values with central tendency like mean, median and mode value of that variable.
5. Various other methods like exploration, regression method etc. has also been used to treat the missing values.

Note: These missing values are represented by dots ('.') for numerical variable and blank for categorical variables in the R.

Extreme value treatment: - This step is done to understand the distribution of our independent variables. Presence of an abnormal value in one of the independent variables can affect the entire analysis (Leverage variable). The extreme values are treated by capping. Capping is required as sometimes, the variable may contain extreme values corresponding to some observations; whereas in reality, such values are unlikely to exist. Such values may be a result of wrong keying the data or may represent the default values. There may be cases of negative values which is logically incorrect for a given variable. In such cases, these values need to be capped because they may affect the mean of the variable drastically.

Some basic rules for capping are as below:

- Don't cap the value unless it is unrealistic
- Cap it to the next highest/lowest (realistic) value.

- Cap at a value so that the continuity is maintained.

Level of the data/roll-up

The sales data can be at the various levels. The main product can have multiple sub products and sales data will need aggregation to compute data at product level. Similarly the transactional sales data for various products are initially at different times and across different shops/channels/regions. This data needs aggregation so that final data is at specific region level and have a natural time span. Eg. Daily, Weekly or Monthly.

The above transformation needs following R procedures:

Use of SQL with count() , sum() and group by functions

Proc transpose

Multicollinearity - When two or more independent variables are related between them, we tell that they have multicollinearity among each other. In technical terms, we say the one variable can be explained as a linear combination of other variables. Multicollinearity among independent variables does not allow the independent variables to explain their impact on the dependent variable optimally due to high internal impact. Keeping collinear variables in the model makes it unstable. In such a scenario we drop one of the variables from the model. Multicollinearity among the variables indicates that these explanatory variables are carrying a common set of information in explaining the dependent variable.

Detection of multicollinearity:

Variance Inflation Factor: We generally test the presence of multicollinearity using Variance Inflation Factor (VIF). Variance Inflation factor (VIF) is obtained by regressing each independent variable, say X on the remaining independent variables (say Y and Z) and checking how much of it (of X) is explained by these variables. Hence VIF = $1 / (1 – R^2)$.

If the VIF value be greater than 2 we drop the variable from the model.

The regression is done iteratively to fix multi-collinearity.

Data Modeling:

Once the data treatment is over we go ahead with the model building;

The modeling is done using proc glm over the Development (Training) and Validation (Testing) Sample:-Before building the model on the data, divide entire data in the ratio of 70:30 as

development sample and validation sample. Development sample is used to develop the model whereas validation sample will be used to check validity of the model. Build the model on development sample data and use the estimates obtained from this model to score the validation sample. If the response captured from the validation sample is nearly equal to the response captured from the development sample then we can say that the model is robust in predicting the responses for future dataset.

Regression Model Building using Proc glm:-Regression technique is used to assess the impact of independent variables over dependent variable. The approach is explained in the following steps

```
*------------------------------------------------------------*;
/*lm function to perform regression*/
*------------------------------------------------------------*;
install.packages(c("car", "sqldf", "plyr"))

fit <- lm(ln_sales ~ cmpgn1 + campgn2 + campgn3 + ln_P_A + ln_P_B + ln_P_C, data=data)

# Prints Output
summary(fit)

#R-square Value
summary(fit)$r.squared

library(car)
## Checks Multicolinearlity
vif(fit)
## to identify Problematic Variables
sqrt(vif(fit)) > 2
```

Regression output:

```
Call:
lm(formula = ln_sales ~ cmpgn1 + campgn2 + campgn3 + ln_P_A +
   ln_P_B + ln_P_C, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.205913 -0.048016  0.006105  0.043470  0.170177
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.12609    0.03239  -3.893 0.000617 ***
cmpgn1       0.13555    0.05206   2.604 0.015036 *
campgn2      0.24318    0.04899   4.964 3.70e-05 ***
campgn3      0.35321    0.04924   7.173 1.28e-07 ***
ln_P_A       0.03772    0.03272   1.153 0.259472
ln_P_B      -0.36176    0.33620  -1.076 0.291806
ln_P_C       0.37920    0.03360  11.284 1.62e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.09872 on 26 degrees of freedom
Multiple R-squared:  0.8649,             Adjusted R-squared:  0.8337
F-statistic: 27.73 on 6 and 26 DF,  p-value: 4.029e-10
```

The analysis and further processing of the regression results is required to generate the output. The initial data taken for regression was mean centered and hence we will get the effect of mean back in the equation.

The code for mean centering fix:

```
/*assigning the mean values*/
## mean centering fix

##assigning the mean values
library(plyr)
##ddply gives summary by group: group on region_cd, keyword summrise
mean1 <- as.data.frame(ddply(data,~Region_cd,summarise,mln_sales=mean(ln_sales)))
mean2 <- as.data.frame(ddply(data,~Region_cd,summarise,mcmpgn1=mean(cmpgn1)))
mean3 <- as.data.frame(ddply(data,~Region_cd,summarise,mcampgn2=mean(campgn2)))
mean4 <- as.data.frame(ddply(data,~Region_cd,summarise,mcampgn3=mean(campgn2)))
mean5 <- as.data.frame(ddply(data,~Region_cd,summarise,mln_P_A=mean(ln_P_A)))
mean6 <- as.data.frame(ddply(data,~Region_cd,summarise,mln_P_B=mean(ln_P_B)))
mean7 <- as.data.frame(ddply(data,~Region_cd,summarise,mln_P_C=mean(ln_P_C)))

/*append means to the dataset*/
##append means to the dataset
## sqldf works only on version 3.1.2 and above
library(sqldf)
```

```
meancnt <- sqldf('select
a.*,b.mln_sales,c.mcmpgn1,d.mcampgn2,e.mcampgn3,f.mln_P_A,g.mln_P_B,h.mln_P_C
            from data a
            left join mean1 b on a.Region_cd = b.Region_cd
            left join mean2 c on a.Region_cd = c.Region_cd
            left join mean3 d on a.Region_cd = d.Region_cd
            left join mean4 e on a.Region_cd = e.Region_cd
            left join mean5 f on a.Region_cd = f.Region_cd
            left join mean6 g on a.Region_cd = g.Region_cd
            left join mean7 h on a.Region_cd = h.Region_cd')


##View and Verify the new merged dataset*/
View(meancnt)
```

The modified and new merged dataset will be used to get the mathematical equation

##Calculation based on the seasonality factor 1.05874 and estimates as derived from model and taking mean centering in consideration

```
meancnt1 <- meancnt

meancnt1$pred = 1.05874*exp(0.1356*(meancnt1$cmpgn1-meancnt1$mcmpgn1)+
0.2432*(meancnt1$campgn2-meancnt1$mcampgn2)+
                0.3532*(meancnt1$campgn3-meancnt1$mcampgn3)+
0.03772*(meancnt1$ln_P_A-meancnt1$mln_P_A)-
                0.3618*(meancnt1$ln_P_B-meancnt1$mln_P_B)+ 0.3792*(meancnt1$ln_P_C-
meancnt1$mln_P_C)+ meancnt1$mln_sales)

meancnt1$res = meancnt1$pred - meancnt1$sales
meancnt1$abs_res = abs(meancnt1$res)
meancnt1$mape = 100*meancnt1$abs_res/meancnt1$sales;

##View and Verify the new dataset with predicted value*/
View(meancnt1)
```

The calculated dataset is used to further develop the Contribution matrix
Contribution matrix is used to evaluate the contribution of the significant factors to total sales. Sale is considered as sum of base sales and fluctuations. The fluctuations are changes from the mean levels of the sales.

The R code for calculating the contributions is as below:

```
/*preparing the contribution matrix */
contribution <- meancnt
attach(contribution)

## Calculate value when all factors are present
contribution$pred = 1.05874*exp(0.1356*(cmpgn1-mcmpgn1)+ 0.2432*(campgn2-
mcampgn2)+ 0.3532*(campgn3-mcampgn3)+ 0.03772*(ln_P_A-mln_P_A)-
0.3618*(ln_P_B-mln_P_B)+ 0.3792*(ln_P_C-mln_P_C)+ mln_sales)
contribution$res = contribution$pred - contribution$sales

## Calculate value when all factors except cmpgn1 are present
contribution$pred_cmpgn1 = 1.05874*exp(0.1356*(-mcmpgn1)+ 0.2432*(campgn2-
mcampgn2)+ 0.3532*(campgn3-mcampgn3)+ 0.03772*(ln_P_A-mln_P_A)-
0.3618*(ln_P_B-mln_P_B)+ 0.3792*(ln_P_C-mln_P_C)+ mln_sales)

## Calculate contribution of cmpgn1 as diff of values when all factors are present
##and when all factors except cmpgn1 are present
contribution$contr_cmpgn1 = contribution$pred - contribution$pred_cmpgn1

contribution$pred_campgn2 = 1.05874*exp(0.1356*(cmpgn1-mcmpgn1)+ 0.2432*(-
mcampgn2)+ 0.3532*(campgn3-mcampgn3)+ 0.03772*(ln_P_A-mln_P_A)-
0.3618*(ln_P_B-mln_P_B)+ 0.3792*(ln_P_C-mln_P_C)+ mln_sales)
contribution$contr_campgn2 = contribution$pred - contribution$pred_campgn2

contribution$pred_campgn3 = 1.05874*exp(0.1356*(cmpgn1-mcmpgn1)+ 0.2432*(campgn2-
mcampgn2)+ 0.3532*(-mcampgn3)+ 0.03772*(ln_P_A-mln_P_A)-
0.3618*(ln_P_B-mln_P_B)+ 0.3792*(ln_P_C-mln_P_C)+ mln_sales)
contribution$contr_campgn3 = contribution$pred - contribution$pred_campgn3;

contribution$pred_ln_P_A = 1.05874*exp(0.1356*(cmpgn1-mcmpgn1)+ 0.2432*(campgn2-
mcampgn2)+ 0.3532*(campgn3-mcampgn3)+ 0.03772*(-mln_P_A)-
0.3618*(ln_P_B-mln_P_B)+ 0.3792*(ln_P_C-mln_P_C)+ mln_sales)
contribution$contr_ln_P_A = contribution$pred - contribution$pred_ln_P_A
```

```
contribution$pred_ln_P_B = 1.05874*exp(0.1356*(cmpgn1-mcmpgn1)+ 0.2432*(campgn2-
mcampgn2)+ 0.3532*(campgn3-mcampgn3)+ 0.03772*(ln_P_A-mln_P_A)- 0.3618*(-mln_P_B)+
0.3792*(ln_P_C-mln_P_C)+ mln_sales)
contribution$contr_ln_P_B = contribution$pred - contribution$pred_ln_P_B

contribution$pred_ln_P_C = 1.05874*exp(0.1356*(cmpgn1-mcmpgn1)+ 0.2432*(campgn2-
mcampgn2)+ 0.3532*(campgn3-mcampgn3)+ 0.03772*(ln_P_A-mln_P_A)- 0.3618*(ln_P_B-
mln_P_B)+ 0.3792*(-mln_P_C)+ mln_sales)
contribution$contr_ln_P_C = contribution$pred - contribution$pred_ln_P_C

#Select Required Variables
contribution <- sqldf('select ln_sales, cmpgn1, campgn2, campgn3, ln_P_A, ln_P_B, ln_P_C,
mln_sales, mcmpgn1, mcampgn2, mcampgn3, mln_P_A, mln_P_B, mln_P_C, timeperiod,
region_cd, pred, sales, res, pred_cmpgn1, pred_campgn2, pred_campgn3, pred_ln_P_A,
pred_ln_P_B, pred_ln_P_C, contr_cmpgn1, contr_campgn2, contr_campgn3, contr_ln_P_A,
contr_ln_P_B, contr_ln_P_C
from contribution')


##Select required variables to generate contribution matrix
cont_mat1 <- sqldf('select pred,contr_cmpgn1,contr_campgn2,contr_campgn3 from
contribution')
write.csv(cont_mat1,file="cont_mat1.csv",row.names=F)
```

The output is finally prepared to showcase the contribution as below: