Marketing Vehicles' Efficiency Analysis

Executive Summary

The executive summary outlines a detailed analysis of the strategic partnership between ACSE Supermarket and Mahou San Miguel, with a focus on promoting and selling San Miguel beer products through various channels. The report uses extensive data preparation and analysis to evaluate the effectiveness of Mahou San Miguel's marketing campaigns and identify promotional activities that generate significant incremental sales.

The data preparation involved aggregating the sales quantity and calculating the average price per product per week from daily observations in the transactions data. Seasonality and holiday data were combined with transaction data using common year_week columns, with dummy variables created for each holiday. Promo data was bifurcated based on product information, and dummy variables were generated for each promotion vehicle for amount and cost data. Reach for each promotion vehicle was determined using adstock values based on GRP values, with alpha values for TV and radio determined using half-lives of 6 weeks and 3 weeks, respectively.

The modeling section involved decomposing weekly sales volume for two San Miguel products into components including Base, Price, Flyer, In-Store Display, Facebook Banner, Google Paid Search, TV, and Radio. The data frames for the two products were used to create two multiplicative models using ridge regression to calculate DueTos for each vehicle. We then Debias DueTos, and use Debiased DueTos to calculate ROI. The final vehicle showing a negative ROI indicating ineffective investment.

Based on the ROI for each activity, we recommend allocating budgets as follows for SAN MIGUEL ESPECIAL 6 PK: Increase the budget for Radio, maintain the same budgets for Google Paid Search, lower the budgets for TV and Flyers, and consider discontinuing Display Advertising and Facebook Banners. For SAN MIGUEL ESPECIAL: Increase the budget for Radio the most, maintain the same budgets for Display Advertising, Flyers, Google Paid Search, and TV. Meanwhile, discontinue promoting using Facebook Banners. By making these adjustments, we believe that Mahou San Miguel can maximize the effectiveness of their marketing activities and optimize their return on investment.

Table of Contents

Introduction

Background

ACSE Supermarket is a renowned retail chain with over 40 stores located throughout Locouria. It boasts an extensive range of over 100,000 products spanning across 100 different categories, catering to the diverse needs and preferences of its customers. One of the primary ways ACSE engages with its customers is through the ACSE Rewards program, where shoppers can take advantage of exclusive weekly deals and promotions.

ACSE has also established strategic partnerships with various suppliers to fund and execute promotional campaigns that drive sales. Promotions have become a significant contributor to the supermarket's sales, and ACSE frequently promotes the beer category in its stores.

Mahou San Miguel, a Spanish brewery that ventured into the Locouria market in 2018, has partnered with ACSE to sell two San Miguel beer products in the retail chain's stores. ACSE and Mahou San Miguel work closely to advertise and promote their products via weekly flyers and in-store displays. In addition to these traditional promotional strategies, Mahou San Miguel also employs various digital marketing techniques such as Facebook banners and Google paid search, as well as traditional media such as television and radio advertisements. With these efforts, Mahou San Miguel aims to enhance its market share and reach a wider audience.

Project Scope

Project Overview:

The objective of the client, Mahou San Miguel, is to assess the efficacy of its marketing collaboration with ACSE and determine the promotional initiatives and marketing strategies that have resulted in notable growth in sales. The client is keen on recognizing the promotion and marketing activities that have contributed to a substantial increase in sales and intends to continue these activities in 2023.

Project Objective:

- Using Market Mix Modeling (MMM) technique to decompose weekly sales volume for each San Miguel product into the following components:
    - Base (includes Seasonality and Events/Holidays)
    - Price

- Flyer
- In-Store Display
- Facebook Banner
- Google Paid Search
- TV
- Radio

- Make recommendations to Mahou San Miguel on which promo for each product and marketing activities to continue in 2023

Data Understanding

Dataset column names:

1. **transactions** contains supplementary transaction history for San Miguel products in 2019 and 2020
2. **products** contains supplementary records for San Miguel products
3. **promo_ad** contains the promotion and advertising/marketing activity records with the following fields:
   a. week – the date of the Sunday representing the start of the week
   b. vehicle – the promo or advertising vehicle
   c. amount – the amount of advertising in units; a "1" means the promotion (Flyer or In-Store Display) is on for the week
   d. unit – the unit of advertisement indicated by the amount
   e. prod_id – the Product ID of the product featured in the promo or advertisement; "ALL" means all San Miguel products
4. **seasonality** contains the seasonality index for products in the Beer category
5. **event_holiday** contains a list of national events and holidays celebrated in Locouria in 2019 and 2020

Data Preparation

The Data Preparation started with loading the relevant excel data into our python environment where we splitted the single file and stored each sheet within that file as a separate data frame which resulted in us ending up with 5 separate data frames capturing the information pertaining to Transactions, Products, Seasonality, Promos and Holidays Respectively.

The next step was to make sure that we have consistency across all our data frames in order to carry out our intended analysis. Upon a closer inspection we realized that there was some inconsistency regarding the nature of data we had at hand. E.g. the data set related to transactions contained daily observations whereas the seasonality, holidays and promos data frames were weekly based as opposed to daily based. Therefore, we thought converting the daily transactions data into weekly transactions would be the ideal way to go about it. In order to achieve that a new column was added to each data frame called year_week that captured the year and week for each observation across all the data sets.

Secondly, to ensure that we only have weekly data for transactions instead of daily observations we aggregated the Sales quantity over each year_week to find total sales quantity and calculated the average price for each product for each week of each year after creating a new column called price which was the result of dividing sales amount and sales quantity. These two steps not only ensured that we had a common column across each datasets that can be used to join all of the datasets but it also ensured that we have weekly data for each of the entities which would later on become the basis of our entire analysis. As a result our transactions data frame looked like this:

	year_week	prod_id	sales_qty	price
0	2019-04	23141592	1886	3.9900
1	2019-04	23141593	2465	9.9900
2	2019-05	23141592	777	4.9900
3	2019-05	23141593	655	13.9900

The next step was to merge the seasonality and holidays data with the transactions data. For seasonality the formatting was ideal so It was merged with the transactions data based on the common year_week column. However, the holidays data needed some transformation before it could be joined. Since, the holiday dataset had one observation per holiday but we had all weeks of data in our transactions data we could either transport it as one column and populate the respective holidays against the corresponding year_week and for the rest of the weeks assign a null value or we could make dummy variables for each holiday so that It gets converted into a set of binary columns using one hot encoding. We chose to go with the second approach as we thought from the modeling perspective having a separate variable for each holiday allows us to gauge the impact of each holiday on our target variable more effectively and it's also more interpretable compared to a single categorical column.

After the one hot encoding the holidays data was merged with the previously merged

year_week	prod_id	sales_qty	price	seas_index	Christmas	Easter	Halloween	Labor Day	Memorial Day	National Day	New Year	Pre Easter	Pre Super Bowl	Super Bowl
2019-04	23141592	1886	3.9900	0.8706	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
2019-05	23141592	777	4.9900	0.9275	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
2019-06	23141592	576	4.9900	0.8830	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

transactions and seasonality data. The result was the data frame shown below:

Before the promo data could be joined with the above data we thought of splitting the data based on product information as we realized that the promo dataset had information which was applicable to only one product in most cases where a given promotion was only applied to one of the two products. Keeping product-specific information in the combined data set seemed like an unideal approach hence we decided to go with separate datasets for each product.

Afterwards, the promo dataset was split into two separate datasets, each representing promo data specific to that product. For observations that were applicable to both products, we decided to divide the Total cost incurred for a particular vehicle in a given week over both products equally which resulted in that observation/promotion being a part of both the product-specific promo datasets but the cost was 50% compared to the original promo dataset as we divided the total cost over the two products. The resulting data frame for each product has been shared below

	week	vehicle	amount	unit	prod_id	cost	year_week
0	2019-01-27	Display	1	NaN	ALL	2250.0000	2019-04
1	2019-03-03	Display	1	NaN	23141593	3000.0000	2019-09
2	2019-03-10	Display	1	NaN	23141593	3000.0000	2019-10
3	2019-03-31	Display	1	NaN	23141593	3000.0000	2019-13
4	2019-04-07	Display	1	NaN	23141593	3000.0000	2019-14

For the same reason that we pivoted the holiday data we decided to pivot both the promo tables such that we end up with dummy variables for each Vehicle and as each vehicle was associated with a corresponding amount and cost we ended up with two sets of dummy variables. One set representing the respective amount of each vehicle for a given time period and the other set representing the respective costs associated with each vehicle for a given period of time. The results have been attached below:

year_week_	Display_Amount	Facebook_Banner_Amount	Flyer_Amount	Google_Paid_Search_Amount	Radio_Amount
2019-01	0.0000	24627.0000	0.0000	2357.0000	0.0000
2019-02	0.0000	43313.0000	0.0000	8734.0000	0.0000

TV_Amount	Display_Cost	Facebook_Banner_Cost	Flyer_Cost	Google_Paid_Search_Cost	Radio_Cost	TV_Cost
30.0000	0.0000	490.3750	0.0000	222.5000	0.0000	8000.0000
15.0000	0.0000	857.6250	0.0000	930.0000	0.0000	4000.0000

The next part was to combine the promo dataset with the merged dataset we created earlier so we ended up with a dataset of weekly transactions, seasonality, holidays, and promo data at the end.

Reach Calculation:

Once our dataset was ready our next task was to calculate the reach for each vehicle. Since we had a total of 6 promotion vehicles but only 3 units which were binary indicator (1/0) of whether a promotion was displayed or not, total impressions and GRP. We decided to keep the Total impressions and binary indicator as it is and considered them to be a good representative of reach whereas for Radio and TV where the unit was GRP we decided to convert the GRPs to reach.

The first step was the calculation of Adstocks using the GRP values for which we used the given formula below. We first calculated the alphas for both TV and Radio based on their half lives of

Ad Stock – model natural short-term retention of media effect on consumers over time:

$$AdStock_t = \alpha GRP_t + (1-\alpha)AdStock_{t-1}$$

$$\sum_t AdStock_t = \sum_t GRP_t$$

The decay parameter, α, associated with a half-life, h, is computed as:

$$\alpha = 1 - (0.5)^\wedge(1/h)$$

6-week and 3-week respectively and the values came out to be **0.109** for TV and **0.206** for Radio.

After the calculation of alpha values, the next step was to use them with the respective GRP values to calculate the Adstock values for each week or time period. We did that using the above formula which required us to take the product of the alpha value and the GRP of a given time period and then add them to the product of the previous period's adstock and (1- alpha).

After the implementation of our code for the above formula we ended up with 4 new columns. temp _tv, temp_radio, ads_tv and ads_radio. The temp variables are basically the product of the alpha value and the GRP for a given period for a given vehicle whereas the latter two variables are the Adstock values for that particular week for a given vehicle and those are calculated using the temp variables and the adstock values of the previous period as depicted in the above formula too. Attached below is a code snippet that performs this entire process from the start.

GRP > AdStock GRP

```
h_tv = 6
h_radio = 3
alpha_tv = 1- pow(0.5, (1/h_tv)) # 0.109
alpha_radio = 1- pow(0.5, (1/h_radio)) # 0.206
```

```
ReachDF=All93df
#calculate row 1 for adstock_tv & adstock_radio
#initilize new column
ReachDF['temp_radio'] = 0
ReachDF['temp_tv'] = 0
#initilize value in first row for both new column
ReachDF.at[0, 'temp_radio'] = ReachDF.at[0, 'Radio_Amount'] * alpha_radio
ReachDF.at[0, 'temp_tv'] = ReachDF.at[0, 'TV_Amount'] * alpha_tv
```

```
#compute alpha*GRP at week T
ReachDF['temp_radio'].iloc[1:] = ReachDF['Radio_Amount'].iloc[1:]* alpha_radio
ReachDF['temp_tv'].iloc[1:] = ReachDF['TV_Amount'].iloc[1:]* alpha_tv
```

```
#initilize new column
ReachDF['ads_radio'] = 0
ReachDF['ads_tv'] = 0
#initilize value in first row for both new column
ReachDF.at[0, 'ads_radio'] = ReachDF.at[0, 'temp_radio']
ReachDF.at[0, 'ads_tv'] = ReachDF.at[0, 'temp_tv']
```

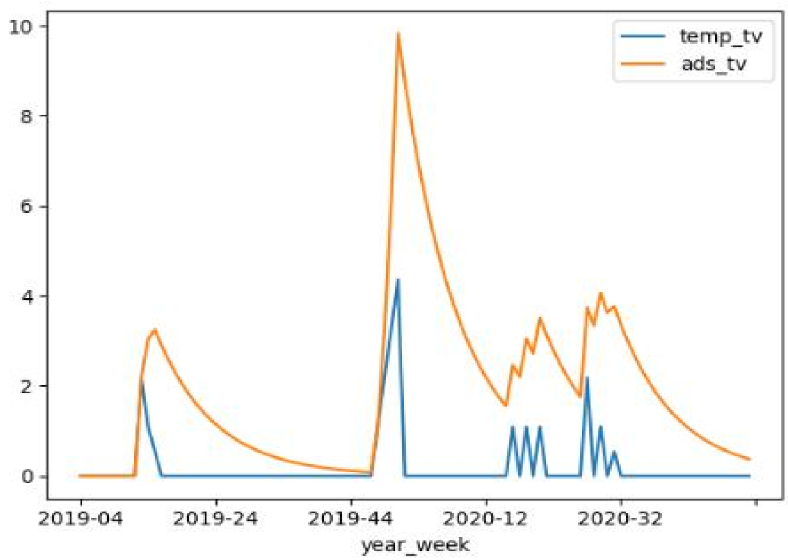
```
#add (1-alpha)*Adstock at week T-1.
# Radio: (1-alpha_radio= 0.794)
# TV: (1-alpha_TV= 0.891)
for i in range(1, len(ReachDF)):
    #Radio
    ReachDF.loc[i, 'ads_radio'] = (1-alpha_radio) * ReachDF.loc[i-1, 'ads_radio']
    ReachDF.loc[i, 'ads_radio'] = ReachDF.loc[i, 'temp_radio'] + ReachDF.loc[i, 'ads_radio']
    #TV
    ReachDF.loc[i, 'ads_tv'] = (1-alpha_tv) * ReachDF.loc[i-1, 'ads_tv']
    ReachDF.loc[i, 'ads_tv'] = ReachDF.loc[i, 'temp_tv'] + ReachDF.loc[i, 'ads_tv']

ReachDF[['Radio_Amount', 'temp_radio','ads_radio','TV_Amount', 'temp_tv','ads_tv']].head(150)
```

	Radio_Amount	temp_radio	ads_radio	TV_Amount	temp_tv	ads_tv
0	10.0000	2.0630	2.0630	0.0000	0.0000	0.0000
1	5.0000	1.0315	2.6689	0.0000	0.0000	0.0000
2	0.0000	0.0000	2.1183	0.0000	0.0000	0.0000
3	0.0000	0.0000	1.6813	0.0000	0.0000	0.0000
4	0.0000	0.0000	1.3344	0.0000	0.0000	0.0000
5	0.0000	0.0000	1.0592	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.8406	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.6672	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.5296	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.4203	20.0000	2.1820	2.1820
10	0.0000	0.0000	0.3336	10.0000	1.0910	3.0350

To validate our results we did a time series analysis for the GRP and their respective adstock values. Since, the more GRPs an advertisement has, the more adstock it is likely to generate so we were expecting a similar trend based on our calculations which were proven right by the time series analysis given below. The blue line is the GRP whereas the yellow line is the Adstock associated with it. We can clearly see the relationship between the two: the increase or decrease in GRP is accompanied by an increase or decrease in Adstock respectively.

```
[ ] #validation using graph--TV
    ReachDF.plot(x='year_week', y=['temp_tv', 'ads_tv'])
    plt.show()
```



The last step was the calculation of reach values from the adstock values calculated in the last part. For that we used the below given formulas for TV and Radio respectively. The adstock values were plugged in place of the GRP variable as depicted in the formula and then the rest of the equation was solved to come up with the reach values for each week and each vehicle for each product.

- TV advertisements have a 6-week half-life, and GRPs map into target audience 2+ Reach via the following formula

$$Reach \ = \ 0.95 \left(1 - e^{-0.40 \ GRP}\right)$$

- Radio advertisements have a 3-week half-life, and GRPs map into target audience 2+ Reach via the following formula

$$Reach \ = \ 0.90 \left(1 - e^{-0.50 \ GRP}\right)$$

The output after the calculation of reach values looked something like this:

AdStock GRP > Reach

```
import math
# define the function to calculate Reach based on Ads_stock
def calculate_reach_Radio(x):
    return 0.9 * (1 - math.exp(-0.5 * x))
# apply the function to radio
ReachDF['reach_radio'] = ReachDF['ads_radio'].apply(calculate_reach_Radio)
ReachDF[['Radio_Amount', 'temp_radio','ads_radio','reach_radio','TV_Amount', 'temp_tv','ads_tv']].head(150)
```

	Radio_Amount	temp_radio	ads_radio	reach_radio	TV_Amount	temp_tv	ads_tv
0	10.0000	2.0630	2.0630	0.5792	0.0000	0.0000	0.0000
1	5.0000	1.0315	2.6689	0.6630	0.0000	0.0000	0.0000
2	0.0000	0.0000	2.1183	0.5879	0.0000	0.0000	0.0000
3	0.0000	0.0000	1.6813	0.5117	0.0000	0.0000	0.0000
4	0.0000	0.0000	1.3344	0.4382	0.0000	0.0000	0.0000
5	0.0000	0.0000	1.0592	0.3700	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.8406	0.3089	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.6672	0.2553	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.5296	0.2094	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.4203	0.1706	20.0000	2.1820	2.1820
10	0.0000	0.0000	0.3336	0.1383	10.0000	1.0910	3.0350

So, the data preparation part was concluded with the calculation of the reach for each vehicle. We ended up with the required dataset for our modeling and the calculation of the Return on Investment for each vehicle. The final dataset after data preparation contained weekly transactions data, seasonality, Dummy variables for holidays and Individual variables of Amount and cost for each vehicle, reach values and we had two such datasets one for each product.

Modeling

The modeling section of our project consists of decomposing the weekly sales volume for each San Miguel products (23141592 and 23141593) into the following components:

- Base (including Seasonality and Events/Holidays)
- Price
- Flyer
- In-Store Display
- Facebook Banner
- Google Paid Search
- TV
- Radio

To achieve the latter goal, we will use the two data frames (one for each of the San Miguel products) and create two multiplicative models (one for each product).

The column names in each of the data frames are the same, and consist of the following:

```
df.columns
✓ 0.0s

Index(['year_week', 'prod_id', 'sales_qty', 'price', 'seas_index', 'Christmas',
       'Easter', 'Halloween', 'Labor Day', 'Memorial Day', 'National Day',
       'New Year', 'Pre Easter', 'Pre Super Bowl', 'Super Bowl',
       'Display_Amount', 'Facebook_Banner_Amount', 'Flyer_Amount',
       'Google_Paid_Search_Amount', 'Radio_Amount', 'TV_Amount',
       'Display_Cost', 'Facebook_Banner_Cost', 'Flyer_Cost',
       'Google_Paid_Search_Cost', 'Radio_Cost', 'TV_Cost', 'reach_radio',
       'reach_tv'],
      dtype='object')
```

We duplicated the two data frames we generated to facilitate ROI calculations and future computations. Moreover, we extracted a subset of the original data frames by removing the 'year_week' and 'prod_id' columns, as they are unnecessary for creating multiplicative models. We took the logarithm of the target variable, i.e., sales quantity, while retaining the independent features in their original form. One benefit of logging the target variable is that it boosts the R-squared of the model.

We are adhering to the following multiplicative model formula:

$$\log(y_t) = \sum_i \beta_i g_i(x_{it}) + \epsilon_t,$$

Moving on to the model building part, we will use a ridge regression model. The idea behind using this model is that regularization will happen by avoiding having a value of 0 for some of our coefficients, but instead to have a very close to 0 value.

```
# model fo prod 93
alpha = 1
ridge_lr = Ridge(alpha=alpha)
ridge_lr.fit(X, y)
y_pred = ridge_lr.predict(X)
r2 = r2_score(y, y_pred)
coefficients = ridge_lr.coef_.flatten()
intercept = ridge_lr.intercept_.flatten()
cocept = np.concatenate((coefficients, intercept))
```

The alpha represents the balance we chose in order for our model not to overfit nor underfit our training data. As seen above, we fit our model, retrieve the R-squared as the evaluation metric and concatenate all the coefficients and the intercept (in this order) in the "cocept" table. The latter process is done twice, for each of the San Miguel products.

```
price: -0.294347573558565
seas_index: 0.6193939750483466
Christmas: 0.21429150262580934
Easter: -0.019103594769539184
Halloween: 0.05424964981692109
Labor Day: 0.003244644313918274
Memorial Day: -0.014553705181494559
National Day: 0.21552532504663205
New Year: 0.05091778064994379
Pre Easter: -0.01435274681203924
Pre Super Bowl: 0.10016691882013228
Super Bowl: 0.09900776895888573
Display_Amount: 0.15791253009845205
Facebook_Banner_Amount: -8.49048397301774e-07
Flyer_Amount: 0.4275844188578135
Google_Paid_Search_Amount: 3.766309131842853e-05
reach_radio: 0.34225094931591066
reach_tv: 0.4157289583783569
intercept: 6.894816941776522

R-squared score for Ridge regression with alpha=1: 0.9581
RMSE: 0.10890508821602601
```

As we can see above, here are the coefficients for the first San Miguel product with ID 23141593. The R-squared recorded is fairly high (95%) with a low RMSE between actual and predicted which is an indicator of how good the ridge regression was able to fit our data.

```
price: -0.11325817094752173
seas_index: 0.4984688327174216
Christmas: 0.9509030100767761
Easter: -0.07773973210225507
Halloween: 0.30510650670949074
Labor Day: 0.04900597189846798
Memorial Day: 0.009586716272634958
National Day: 0.2832497197742374
New Year: 0.26410255504631025
Pre Easter: 0.1891619130887243
Pre Super Bowl: 0.5771468130097842
Super Bowl: 0.44041747397358466
Display_Amount: 0.32288839999093094
Facebook_Banner_Amount: 3.235173110446491e-08
Flyer_Amount: 0.32609405841709155
Google_Paid_Search_Amount: 0.0001232449951846596
reach_radio: 0.5476260429454612
reach_tv: 0.7849884213812234
intercept: 5.469097352287304
R-squared score for Ridge regression with alpha=1: 0.9419
RMSE: 0.22157335448230153
```

The aforementioned procedure was replicated on the training data for another San Miguel product, identified by the ID 23141592. Although the coefficients are identical, the RMSE for this product is approximately twice as high as the previous one.

To calculate the DueTos for multiplicative models for each marketing vehicle, we will employ the following formula and consider the following vehicles:

1. Compute $\mathrm{DueTo}_{it} = \hat{y}_t - \hat{y}_t(x_{it} = x_{itBASE})$
2. Debias: adjust DueTo_{it} by scaling, so that $\sum_i \mathrm{DueTo}_{it} = y_t$

```
lst = ['base','price','Display_Amount','Facebook_Banner_Amount','Flyer_Amount','Google_Paid_Search_Amount',
       'reach_radio', 'reach_tv']

dfs = [df, df_92]
```

To calculate the difference in predicted sales quantity, we must subtract the forecasted sales quantity under base conditions (i.e., no holidays considered). To achieve this, we will set the value of each base variable to 0 and generate a new temporary data frame.

```
base_vars = ['seas_index', 'Christmas', 'Easter', 'Halloween',
        'Labor Day', 'Memorial Day', 'National Day', 'New Year', 'Pre Easter',
        'Pre Super Bowl', 'Super Bowl']
```

The temporary data frame will be employed to forecast the sales quantity and estimate the coefficients for the marketing vehicles when there are no holidays. This provides a baseline sales quantity, utilizing the previously developed ridge regression models (ridge_lr) for both products. To accurately compute the DueTos, we must:

- Apply the exponential function to the sales quantity target variable, as it was previously logged in the model.
- Deduct the predicted sales target, based solely on the base coefficients (i.e., 0 for all holidays), from the forecasted sales target utilizing coefficients from the full dataset that includes holidays.
- Scale the DueTos (i.e., exponent of the base sales quantity divided by the exponent of the predicted non-base sales quantity) to enable comparison on the same scale.
- Sum the DueTos to determine the actual value for each marketing vehicle.

```
dct = {}
l = 0
base_vars = ['seas_index', 'Christmas', 'Easter', 'Halloween',
        'Labor Day', 'Memorial Day', 'National Day', 'New Year', 'Pre Easter',
        'Pre Super Bowl', 'Super Bowl']
for idx, z in enumerate(dfs):
    for c in lst:
        temp_df = z.copy()
        if c == "base":
            temp_df[base_vars] = 0
        elif c != "price":
            temp_df[c] = 0
        else:
            temp_df[c] = temp_df[c].mean()
        Xtemp = temp_df.iloc[:, 1:].values
        if l < 8:
            y_temp = ridge_lr.predict(Xtemp)
            due_to_temp = np.exp(y) - np.exp(y_temp)
            ratio_temp = np.exp(y_temp) / np.exp(y)
            sum_scaled_due_to_temp = np.sum(ratio_temp * due_to_temp)
            dct[(f'df_{idx}', c)] = sum_scaled_due_to_temp
        else:
            y_temp = ridge_lr_92.predict(Xtemp)
            due_to_temp = np.exp(y_92) - np.exp(y_temp)
            ratio_temp = np.exp(y_temp) / np.exp(y_92)
            sum_scaled_due_to_temp = np.sum(ratio_temp * due_to_temp)
            dct[(f'df_{idx}', c)] = sum_scaled_due_to_temp
        l += 1
```

The variable "l" acts as a counter to signal the for loop when to commence computing the DueTos for the second product. The list "dct" will store the DueTos for both products, with df_0 assigned index 0 for product 23141593 and index 1 for 23141592.

Subsequently, we need to determine the ROI for every marketing vehicle by dividing its amount by its summed up costs.

```
lst_roi = ['Display_Amount','Facebook_Banner_Amount', 'Flyer_Amount',
           'Google_Paid_Search_Amount','reach_radio', 'reach_tv']
p_roi = ['Display_Cost', 'Facebook_Banner_Cost', 'Flyer_Cost',
       'Google_Paid_Search_Cost', 'Radio_Cost', 'TV_Cost']
```

✓ 0.0s

```
dct_roi = {}
cost_dfs = [df_cost, df_92_cost]
for idx, cost_df in zip(range(2), cost_dfs):
    for c,p in zip(lst_roi,p_roi):
        roi_temp = dct[(f'df_{idx}', c)]/cost_df[p].sum()
        dct_roi[(f'df_{idx}', c)] = roi_temp
```

```
{('df_0', 'reach_radio'): 0.4381154620964385,
 ('df_0', 'Display_Amount'): 0.30305179351938816,
 ('df_0', 'Flyer_Amount'): 0.2807220281385026,
 ('df_1', 'reach_radio'): 0.26950702296567014,
 ('df_0', 'Google_Paid_Search_Amount'): 0.25676866064642634,
 ('df_0', 'reach_tv'): 0.22371233737466156,
 ('df_1', 'Google_Paid_Search_Amount'): 0.18876872089014424,
 ('df_1', 'reach_tv'): 0.1375968192323545,
 ('df_1', 'Flyer_Amount'): 0.12100976051370985,
 ('df_1', 'Display_Amount'): 0.046756659935707934,
 ('df_1', 'Facebook_Banner_Amount'): 0.007641765703744033,
 ('df_0', 'Facebook_Banner_Amount'): -0.022021513652912477}
```

The ROI for each marketing vehicle is visible above, ranging from the highest return on investment to the lowest. The final vehicle appears to have a negative ROI, indicating that our investment was not utilized effectively and resulted in lower sales.

Recommendation

```
{('df_1', 'reach_radio'): 0.43811546209643853,
 ('df_1', 'Display_Amount'): 0.3030517935193938,
 ('df_1', 'Flyer_Amount'): 0.28072202813850294,
 ('df_0', 'reach_radio'): 0.26950702296566975,
 ('df_1', 'Google_Paid_Search_Amount'): 0.2567686606464265,
 ('df_1', 'reach_tv'): 0.22371233737466162,
 ('df_0', 'Google_Paid_Search_Amount'): 0.1887687208901441,
 ('df_0', 'reach_tv'): 0.1375968192323545,
 ('df_0', 'Flyer_Amount'): 0.1210097605137095,
 ('df_0', 'Display_Amount'): 0.046756659935707795,
 ('df_0', 'Facebook_Banner_Amount'): 0.007641765703743588,
 ('df_1', 'Facebook_Banner_Amount'): -0.022021513652912012}
```

Here are our recommendations to Mahou San Miguel on promotion for both products and marketing activities to continue in 2023

For SAN MIGUEL ESPECIAL 6 PK (df_0) Product ID: 23141593

- **Radio:** It has the greatest ROI among all marketing channels for this particular product. This means that for every dollar spent on Radio advertising, the product generates \$0.2695 in sales. So, with a comparable budget, we recommend continuing the marketing campaign in 2023 and potentially increasing the investment ratio.
- **Google Paid Search:** With a ROI of close to 0.2, Google Paid Search investment has a mediocre effect on sales on this product. As a result, we advise continuing this marketing endeavor in 2023 with a comparable expenditure.
- **TV and Flyers** : TV and Flyer have ROI close to 0.1, which indicate they have relatively little impact on sales. It may be worthwhile for Mahou San Miguel to consider lowering the budget allocated to these activities or finding ways to make them more effective.
- **Display Advertising and Facebook Banners:** they both have an ROI close to 0, indicating that investments in these promotion channels have almost no impact on sales. Mahou San Miguel should consider discontinuing these promotions.

For SAN MIGUEL ESPECIAL (df_1) Product ID: 23141592

- **Radio:** According to the data, with a ROI of 0.4381, Radio has the greatest influence on this product's sales. So, with a bigger budget, we strongly advise continuing this marketing campaign in 2023.
- **Display Advertising, Flyers, Google Paid Search and TV:** With a ROI of greater than 0.2, display advertising moderately affects sales. As a result, we advise continuing this marketing endeavor in 2023 with a comparable expenditure.
- **Facebook Banners** has a negative ROI,which means that it is actually hurting sales rather than helping them. Mahou San Miguel should discontinue promoting using Facebook Banners .

In summary, we recommend allocating budgets as follows:

For SAN MIGUEL ESPECIAL 6 PK:

Increase the budget for Radio, maintain the same budgets for Google Paid Search, lower the budgets for TV and Flyers, and consider discontinuing Display Advertising and Facebook Banners.

For SAN MIGUEL ESPECIAL:

Increase the budget for Radio the most, maintain the same budgets for Display Advertising, Flyers, Google Paid Search, and TV. Meanwhile, discontinue promoting using Facebook Banners.

By making the above adjustments, we believe that Mahou San Miguel can maximize the effectiveness of their marketing activities and optimize their return on investment.