Optimization Project 1 - Linear Programming

Group 4 - Praneet Kumar Alamuri (pa2222), Meeth Yogesh Handa (mh58668), Sai Bhargav Tetali (srt2578), James Anderson (ja47823)

Problem Statement

As per a recent CMO survey, marketing budgets comprise 11 percent of total company budgets. However, the effectiveness of marketing and ROI generated vary across companies. One key reason for this could be the allocation of marketing spend in each advertising/ marketing channel.

In this study, we are using Linear Programming to find the optimal allocation of the company's marketing budget across 10 channels to optimize expected profits based on the ROI estimates provided by 2 consulting firms.

Additionally, based on our company's CMO's experience in the industry, we have been asked to apply the following three constraints to our analysis -

1. The amount invested in print and TV should be no more than the amount spent on Facebook and Email
2. The total amount used in social media (Facebook, LinkedIn, Instagram, Snapchat, and Twitter) should be at least twice of SEO and AdWords
3. For each platform, the amount invested should be no more than $3M

Also, we have a marketing budget of $10 Million decided, which will be the fourth constraint in our optimization problem.

As per the constraints stated above, we created an objective function for the ROI to be maximized by Gurobi with the amount invested in each marketing channel as the different variables.

```
ojModel = gp.Model() # initialize an empty model
# Constraint c: For each platform, the amount invested should be no more than $3m
ojModX = ojModel.addMVar(len(df.columns)-1, ub=np.ones(len(df.columns)-1)*3000000) #initialize x variables and give upper bound
```

```
ojModCon = ojModel.addMConstrs(A, ojModX, sense, b) # add the constraints to the model
ojModel.setMObjective(None,obj,0,sense=gp.GRB.MAXIMIZE)
ojModel.Params.OutputFlag = 0
ojModel.Params.TimeLimit = 3600
ojModel.optimize() # solve the LP
```

After obtaining the optimal value from Gurobi, we print out the value of all the variables determined for the optimization and receive our optimal budget allocation. We can see that if the ROI data from Company 1 is correct, we should be investing the maximum amount of $3 million in TV, Instagram and Email marketing channels plus an additional $1 million in AdWords to maximize our ROI.

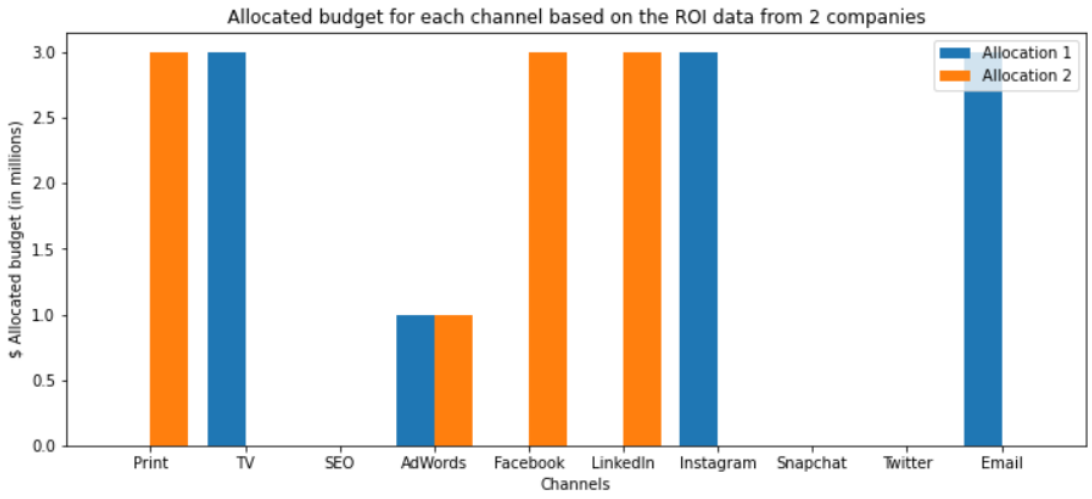
	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email	Total_RoI
Case 1 - All constraints and RoI 1	0.0	3000000.0	0.0	1000000.0	0.0	0.0	3000000.0	0.0	0.0	3000000.0	456000.0

We then modified the objective function after the boss' recommendations of verifying the output with the ROI data received from the second company. The constraints for the optimization problem remain the same.

```
# solving the same LP but this time considering the second RoI data
ojModel_2 = gp.Model() # initialize an empty model
ojModX_2 = ojModel_2.addMVar(len(df.columns)-1, ub=np.ones(len(df.columns)-1)*3000000)
ojModCon_2 = ojModel_2.addMConstrs(A, ojModX_2, sense, b)
ojModel_2.setMObjective(None,obj2,0,sense=gp.GRB.MAXIMIZE)
ojModel_2.Params.OutputFlag = 0
ojModel_2.Params.TimeLimit = 3600
ojModel_2.optimize() # solve the LP
```

On solving the optimization problem, we get the following output and plot it in terms of bar graph for easy interpretation of the differences in budget for each channel:

	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email	Total_RoI
Case 1 - All constraints and RoI 1	0.0	3000000.0	0.0	1000000.0	0.0	0.0	3000000.0	0.0	0.0	3000000.0	456000.0
Case 2 - All constraints and RoI 2	3000000.0	0.0	0.0	1000000.0	3000000.0	3000000.0	0.0	0.0	0.0	0.0	456000.0



As we can see above, the ideal allocation changes if the second ROI data is to be trusted with $3

million being invested into Print, Facebook, LinkedIn each and $1 million to be invested in AdWords.

Testing expected ROI models with opposite optimal allocation:
We tested the expected ROI by marketing type of the first model with the optimal allocation of the second model. If we utilized this method, our ROI would be $252000.

```
In [28]: print("The RoI we get if first RoI data is correct but we used second allocation:",ojModX_2.x @ obj)
         The RoI we get if first RoI data is correct but we used second allocation: 252000.0
```

We also tested the expected ROI by marketing type of the second model with the optimal allocation of the first model. If we utilized this method, our ROI would be $264000.

```
In [29]: print("The RoI we get if second RoI data is correct but we used first allocation:",ojModX.x @ obj2 )
         The RoI we get if second RoI data is correct but we used first allocation: 264000.0
```

These two values are lower than the respective ROIs found above by $204000 and $192000. The obvious conclusion to make is that an optimization model needs to be run every time the expected ROI by marketing type changes.

```
In [42]: print("If first RoI data is correct but we use the second allocations we end up with lower amount by the following va
         If first RoI data is correct but we use the second allocations we end up with lower amount by the following value:
         204000.0

In [43]: print("If second RoI data is correct but we use the first allocations we end up with lower amount by the following va
         If second RoI data is correct but we use the first allocations we end up with lower amount by the following value:
         192000.0
```

The uncertainty in the ROI values calls for the third constraint. In this problem, we explored what would happen if the expected ROI data is incorrect. If we completely removed the third constraint, the first expected ROI model would result in us investing all $10M in TV and Email advertisement. However, if the expected ROI numbers by marketing type were actually found to be closer to the second firm's expected ROI, we will have lost opportunity.
Conceptually, the third constraint also makes sense. Although the $3M is rather arbitrary, the limit implies the diversification of a marketing portfolio; allowing a company to at least reach a broader audience. For example, elderly people may be reached by TV ads while younger people can be reached by Instagram or other social media. Just because advertising towards elderly people may be more profitable now does not mean we do not want to introduce the company / product to the younger generation. This introduction will smooth marketing procedures in the future if a new demographic is to be marketed towards.

We ran a sensitivity analysis to see the upper and lower bounds at which the optimal solution remains the same.

In [32]:
```
print("Lower bounds of the first RoI data which will still result in the first allocation:",ojModX.SAObjLow,"\n")

for i in range(0,len(df_dist.columns)-1):
    variance = obj[i] - ojModX.SAObjLow[i]
    print(df_dist.columns[i],"can decrease by",variance,"without changing optimal allocation")
```

```
Lower bounds of the first RoI data which will still result in the first allocation: [ -inf 0.039  -inf 0.033  -inf
-inf 0.039  -inf  -inf 0.029]

Print can decrease by inf without changing optimal allocation
TV can decrease by 0.010000000000000002 without changing optimal allocation
SEO can decrease by inf without changing optimal allocation
AdWords can decrease by 0.005999999999999998 without changing optimal allocation
Facebook can decrease by inf without changing optimal allocation
LinkedIn can decrease by inf without changing optimal allocation
Instagram can decrease by 0.006999999999999999 without changing optimal allocation
Snapchat can decrease by inf without changing optimal allocation
Twitter can decrease by inf without changing optimal allocation
Email can decrease by 0.015 without changing optimal allocation
```

In [33]:
```
print("Upper bounds of the first RoI data which will still result in the first allocation:",ojModX.SAObjUp,"\n")

for i in range(0,len(df_dist.columns)-1):
    variance = ojModX.SAObjUp[i] - obj[i]
    print(df_dist.columns[i],"can increase by",variance,"without changing optimal allocation")
```

```
Upper bounds of the first RoI data which will still result in the first allocation: [0.049 0.062 0.039 0.046 0.029 0
.039   inf 0.039 0.039   inf]

Print can increase by 0.018000000000000002 without changing optimal allocation
TV can increase by 0.012999999999999998 without changing optimal allocation
SEO can increase by 0.015 without changing optimal allocation
AdWords can increase by 0.006999999999999999 without changing optimal allocation
Facebook can increase by 0.012999999999999998 without changing optimal allocation
LinkedIn can increase by 0.015 without changing optimal allocation
Instagram can increase by inf without changing optimal allocation
Snapchat can increase by 0.013000000000000001 without changing optimal allocation
Twitter can increase by 0.005999999999999998 without changing optimal allocation
Email can increase by inf without changing optimal allocation
```

Allocations based on a full year RoI data:

We have the monthly Return on Investments (RoI) percentages for each of the media for 12 months. Assuming we have 100% confidence on this data we can calculate the optimal allocation for each of the months. Also, since we have permission to reinvest half of the RoI our monthly budget is going to vary every month.

We have two ways in which we can find out the optimal allocations:

1. We take each month individually, find out the optimal allocations by considering the constraints, calculate the RoI for the month and then use that RoI to update our budget for the next month. We use a for loop in python to achieve this. The following is the for loop code chunk with comments:

In [145]:
```
b_mat = np.zeros(3)# for each loop we are going to optimize for only one month for which ther
b_mat[0] = 10000000# budget constraint for the fisrt month

RoI_1 = [] # to store the monthly RoI values
allocation_1 = [] # to store the monthly allocations
investment_1 = [] # to store the monthly budgets

for i in range(months_no):
    ojModel_x = gp.Model() # initialize an empty model
    ojModX_x = ojModel_x.addMVar(channel_no, ub=np.ones(channel_no)*3000000)
    ojModCon_x = ojModel_x.addMConstrs(A, ojModX_x, sense, b_mat)
    ojModel_x.setMObjective(None,obj_mat[i,1:]/100,0,sense=gp.GRB.MAXIMIZE)
    ojModel_x.Params.OutputFlag = 0
    ojModel_x.Params.TimeLimit = 3600
    ojModel_x.optimize()# solve the LP
    investment_1.append(b_mat[0])
    RoI_1.append(ojModel_x.objVal)
    allocation_1.append(ojModX_x.x)
    b_mat[0] = b_mat[0] + ojModel_x.objVal/2 # updating our budget for the next month(assumin
    #b_mat[0] = 10000000 + ojModel_x.objVal/2 # updating our budget for the next month(assumi
```

2. The second way is to consider all the months together which will lead to us having one Linear Problem with (12 months)*(10 media channels) = 120 variables. For this problem, we will have (3 constraints)*(12 months) = 36 constraints. We have 3 constraints for each month. The following is the code chunk in python where we are building our constraints matrix:

In [150]:
```
A_1 = np.zeros((3*months_no,months_no*channel_no)) #120 variables and for each month we have 3 constraints so
# 3x12 = 36 constraints

# using for loops to construct our constraints matrix for each of the month
for i in range(months_no):
    # constraint for total budget
    A_1[i,i*channel_no:i*channel_no+channel_no] = 1
    #constraint for Print and TV less than Facebook and Email
    A_1[i+months_no,i*channel_no+index_no1-1] = 1
    A_1[i+months_no,i*channel_no+index_no2-1] = 1
    A_1[i+months_no,i*channel_no+index_no5-1] = -1
    A_1[i+months_no,i*channel_no+index_no10-1] = -1
    # social media constraint
    A_1[i+2*months_no,i*channel_no+index_no3-1] = -2
    A_1[i+2*months_no,i*channel_no+index_no4-1] = -2
    A_1[i+2*months_no,i*channel_no+index_no5-1] = 1
    A_1[i+2*months_no,i*channel_no+index_no6-1] = 1
    A_1[i+2*months_no,i*channel_no+index_no7-1] = 1
    A_1[i+2*months_no,i*channel_no+index_no8-1] = 1
    A_1[i+2*months_no,i*channel_no+index_no9-1] = 1

for i in range(1,months_no):
    # constraint for total budget
    #This code will enable us to keep the RHS of budget constraint
    #as 10,000,000 by changing our x variables coefficients in the LHS
    A_1[i,i*channel_no-channel_no:i*channel_no] = -0.5*obj_mat[i-1,1:]/100
    # use below if condition code for cumulative.
    if i>=2:
        A_1[i,0:(i-1)*channel_no] = A_1[i-1,0:(i-1)*channel_no]
```

We are using loops to construct our constraint matrix. For the total budget constraint the RHS remains $10M. We are accommodating the extra investment from the previous month's RoI by incorporating the previous month's allocations decision variables in our constraint equation. Please find below March month budget constraint for reference:

```
[-0.02    , -0.018  , -0.012  , -0.0195 , -0.015  , -0.0175 ,
 -0.018   , -0.01125, -0.0175 , -0.0175 , -0.02   , -0.0195 ,
 -0.0135  , -0.019  , -0.0215 , -0.016  , -0.0135 , -0.009  ,
 -0.0185  , -0.0175 ,  1.     ,  1.     ,  1.     ,  1.     ,
  1.      ,  1.     ,  1.     ,  1.     ,  1.     ,  1.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ,
  0.      ,  0.     ,  0.     ,  0.     ,  0.     ,  0.     ],
```

In the above array the first 10 values are half of the RoIs of January month and second 10 values are half of the RoIs of February month. The following 10 1s are the coefficients of March month allocations. The previous month coefficients are negative because when we shift them to the RHS the budget allocation for March month would be $10M+half of previous months RoIs.

The following are the details of Investment, allocations and RoIs for the full year. We need to note that we get the same values by using either of the above mentioned methods.

Month	Investment	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email	RoI
January	$ 10,000,000.00	$ 3,000,000.00	$ -	$ -	$ 1,333,333.00	$ -	$ -	$ 2,666,667.00	$ -	$ -	$ 3,000,000.00	$ 373,000.00
February	$ 10,186,500.00	$ 3,000,000.00	$ -	$ -	$ 2,395,500.00	$ 3,000,000.00	$ -	$ -	$ -	$ 1,791,000.00	$ -	$ 406,296.00
March	$ 10,389,650.00	$ -	$ -	$ -	$ 3,000,000.00	$ -	$ 3,000,000.00	$ 1,389,648.00	$ -	$ 3,000,000.00	$ -	$ 414,416.98
April	$ 10,596,860.00	$ -	$ -	$ -	$ 3,000,000.00	$ -	$ 3,000,000.00	$ 3,000,000.00	$ -	$ 1,596,856.00	$ -	$ 414,486.83
May	$ 10,804,100.00	$ 1,804,100.00	$ -	$ -	$ -	$ -	$ -	$ 3,000,000.00	$ -	$ 3,000,000.00	$ 3,000,000.00	$ 432,143.50
June	$ 11,020,170.00	$ 3,000,000.00	$ -	$ -	$ -	$ -	$ -	$ 3,000,000.00	$ -	$ 2,020,172.00	$ 3,000,000.00	$ 454,766.52
July	$ 11,247,550.00	$ 1,123,777.00	$ -	$ -	$ 3,000,000.00	$ 1,123,777.00	$ -	$ 3,000,000.00	$ -	$ 3,000,000.00	$ -	$ 468,654.64
August	$ 11,481,880.00	$ 3,000,000.00	$ -	$ -	$ 1,827,294.00	$ -	$ 654,588.20	$ -	$ -	$ 3,000,000.00	$ 3,000,000.00	$ 487,966.11
September	$ 11,725,870.00	$ 1,362,933.00	$ -	$ -	$ 3,000,000.00	$ -	$ 3,000,000.00	$ -	$ -	$ 3,000,000.00	$ 1,362,933.00	$ 459,219.95
October	$ 11,955,480.00	$ -	$ -	$ -	$ 3,000,000.00	$ -	$ 3,000,000.00	$ 3,000,000.00	$ -	$ -	$ 2,955,475.00	$ 427,575.21
November	$ 12,169,260.00	$ 3,000,000.00	$ -	$ -	$ 2,056,421.00	$ -	$ 1,112,842.00	$ 3,000,000.00	$ -	$ -	$ 3,000,000.00	$ 517,375.57
December	$ 12,427,950.00	$ 3,000,000.00	$ 3,000,000.00	$ -	$ 427,950.70	$ 3,000,000.00	$ -	$ -	$ -	$ -	$ 3,000,000.00	$ 516,834.17
												$ 5,372,735.48

Allocations based on a full year RoI data but with constraint to have a stable budget:
We can see from the above allocations that we do not have a stable budget. For example, we are not investing in TV from January to November but suddenly in December we are investing $3M. For a stable budget, allocation in a particular media should not deviate by more than $1M. To get allocations such that we have a stable budget we need to follow the second method mentioned above (with 120 variables) and add more constraints to incorporate the difference between monthly allocation of each media. We get an additional 220 constraints on top of the 36

constraints. For each media, in a month we need to add a constraint connecting it to its previous month and next month and also since we are unaware of which value will be higher we need to add 2 constraints with $1M and $-1M as our RHS. In this way for each media we get 22 constraints and for 10 media channels we get 22 * 10 = 220 constraints. In total, we get 220 + 36 = 256 constraints.