Optimization Project Report - Group 1

Apurva Audi, Monica Martinez, Tanushree Devi Balaji, Victor Lai

The primary objective of this report is to find the best marketing budget allocation strategy for this specific organization. While it sounds like a seemingly straightforward narrative, what makes it an optimization worthy problem are the budget constraints that come with the allocation task. This document delineates the approach to answering some of the questions that popped up about budget optimization:

<u>Questions 1 to 3:</u>

1) Assume that your company is deciding how to spend a marketing budget of $10M. You work in the marketing department as a data scientist and the chief marketing officer has asked you to write a report recommending how to spread this budget among several marketing mediums. Your department has employed an outside consulting firm to estimate the return on investment (ROI) of each marketing medium under consideration. The results are in the table below, and also in a CSV attached to this assignment:

Platform	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
ROI	3.1%	4.9%	2.4%	3.9%	1.6%	2.4%	4.6%	2.6%	3.3%	4.4%

2) On top of these ROIs, your boss has decided to constrain your budget as follows:
a. The amount invested in print and TV should be no more than the amount spent on Facebook and Email. Surprisingly, email seems to be a great channel for reaching real people.
b. The total amount used in social media (Facebook, LinkedIn, Instagram, Snapchat, and Twitter) should be at least twice of SEO and AdWords.
c. For each platform, the amount invested should be no more than $3M.

3) Formulate the marketing budget allocation problem as a linear program. Use gurobi to find the optimal budget allocation

Solution:

Let the variables determining allocation for each platform be defined as follows:

Platform	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
Variable	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10

Objective Function: The objective is to maximize the return on investment (ROI). Total value of returns is given by the product of ROI% for that platform and the corresponding allocation.

0.031*x1 + 0.049*x2 + 0.024*x3 + 0.039*x4 + 0.016*x5 + 0.024*x6+ 0.046*x7 + 0.026*x8 + 0.033*x9 + 0.044*x10

```
print(obj1)

[0.031 0.049 0.024 0.039 0.016 0.024 0.046 0.026 0.033 0.044]
```

Constraints:

The constraints are as follows:

1. The total allocation can't exceed 10M
2. Print + TV <= Facebook + Email
3. Facebook + LinkedIn + Instagram + Snapchat + Twitter >= 2 * (AdWords +SEO)
4. Individual allocations can't be greater than 3M
5. Non-negativity constraints

In mathematical terms,

x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 <= 10M

x1 + x2 - x5 - x10 <= 0

x5 + x6 + x7 + x8 + x9 - 2*x4 - 2*x3 >= 0

x1,x2,x3,x4,x5,x6,x7,x8,x9,x10 <= 3M

x1,x2,x3,x4,x5,x6,x7,x8,x9,x10 >= 0

Note that for the rest of the analysis, we shall consider all values to be in millions. Eg: 10M will be denoted as 10 in Python

LHS:

```
A1

array([[ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  0.,  0., -1.,  0.,  0.,  0.,  0., -1.],
       [ 0.,  0., -2., -2.,  1.,  1.,  1.,  1.,  1.,  0.],
       [ 1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  1.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  1.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  1.]])
```

RHS

```
b1

array([[10.],
       [ 0.],
       [ 0.],
       [ 3.],
       [ 3.],
       [ 3.],
       [ 3.],
       [ 3.],
       [ 3.],
       [ 3.],
       [ 3.],
       [ 3.],
       [ 3.]])
```

Running the Model on Gurobi:

Preprocessing:
Import packages and read in necessary files

```
[3] import numpy as np
    import gurobipy as gp
    import pandas as pd
    from google.colab import files
    uploaded = files.upload() #roi_data.csv

    Choose files 2 files
    • ROI_data.csv(text/csv) - 235 bytes, last modified: 27/09/2022 - 100% done
    • roi_mat.csv(text/csv) - 660 bytes, last modified: 27/09/2022 - 100% done
    Saving ROI_data.csv to ROI_data.csv
    Saving roi_mat.csv to roi_mat.csv
```

```
# objective arrays defined here
df = pd.read_csv("ROI_data.csv",index_col="Platform")
obj1 = np.array(df.iloc[0])
obj2 = np.array(df.iloc[1])
print(obj1)
print(obj2)

[0.031 0.049 0.024 0.039 0.016 0.024 0.046 0.026 0.033 0.044]
[0.049 0.023 0.024 0.039 0.044 0.046 0.026 0.019 0.037 0.026]
```

```
[5] # if no csv available (hard coded, do not use for final submission)
    obj1 = np.array([0.031,0.049,0.024,0.039,0.016,0.024,0.046,0.026,0.033,0.044]) # objective vector
    obj2 = np.array([0.049,0.023,0.024,0.039,0.044,0.046,0.026,0.019,0.037,0.026]) # objective vector
    print(obj1)
    print(obj2)

    [0.031 0.049 0.024 0.039 0.016 0.024 0.046 0.026 0.033 0.044]
    [0.049 0.023 0.024 0.039 0.044 0.046 0.026 0.019 0.037 0.026]
```

Modeling Steps:

1. Define objective, constraint & sense matrices such that:
   A * x (<=,=,>=) b and c
   Where A - LHS of constraint matrix, x - variable matrix, (<=,=,>=) - sense, b = RHS of constraint matrix, c - objective matrix

```
# initialize constraint matrix
A1 = np.zeros((13,10))

#Constraints
A1[0,0:10] = 1                            # total investment constraint
A1[1,:] = [1,1,0,0,-1,0,0,0,0,-1]        # constraint in 2.a.
A1[2,:] = [0,0,-2,-2,1,1,1,1,1,0]        # constraint in 2.b.
A1[3:13,0:10] = np.diag(np.ones(10)) # total investment in each channel constraint

# limits on total investment and other constraints for individual investments in each channel
b1 = np.zeros((13,1))
b1[0]=10
b1[3:13]=3

#Inequalities sign for each constraint
sense = np.array(['<','<','>','<','<','<','<','<','<','<','<','<','<'])
```

2. Run the optimization Model on Gurobi

```
BudgetModel1 = gp.Model() # initialize an empty model

BudgetModX1 = BudgetModel1.addMVar(10) # tell the model how many variables there are
# must define the variables before adding constraints because variables go into the constraints
BudgetModCon1 = BudgetModel1.addMConstrs(A1, BudgetModX1, sense, b1) # add the constraints to the model
BudgetModel1.setMObjective(None,obj1,0,sense=gp.GRB.MAXIMIZE) # Ask Gurobi to Maximize the objective

BudgetModel1.Params.OutputFlag = 0
BudgetModel1.Params.TimeLimit = 3600
```

```
Restricted license - for non-production use only - expires 2023-10-25
```

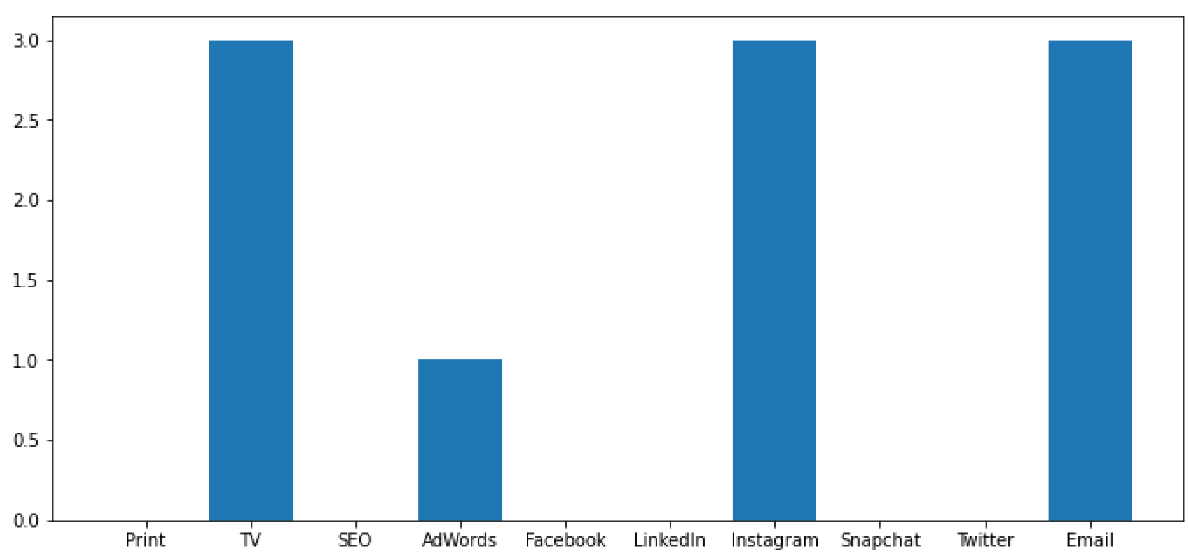
```
[14] BudgetModel1.optimize() # solve the LP
```

```
[15] BudgetModel1.objVal # optimal ROI level

0.45600000000000007
```

```
[16] BudgetModX1.x

array([0., 3., 0., 1., 0., 0., 3., 0., 0., 3.])
```

- Create an empty model
- Add variables, constraints & objective to it and ask Gurobi to maximize the objective
- Obtain optimal point & corresponding objective value

Results of Optimization:
The allocation is as follows (all values are in millions)



Takeaways:
This shows that an investment of 10 million will give an ROI of 0.456 million (i.e) 4.56% when the investment is in the proportions of 0 (print), 3 million (TV), 0 (Seo), 1 million (AdWorks), 0 (Facebook), 0 (LinkedIn), 3 million (Instagram) , 0 (Snapchat), 0 (Twitter) and 3 million (Email).

Question 4:

Your boss is happy to see the promising results presented by the marketing department. However, your boss is also very concerned because your boss recalls being somewhat disappointed after following such recommendations in the past. To be cautious about the decision, your team has decided to get another opinion about the ROI data and rerun the analysis. The second consulting firm returns the estimates of the ROI data in the table below (also in the CSV file mentioned above). You are asked to compare the two optimal allocations from these two ROI estimates.

Platform	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
ROI	4.9%	2.3%	2.4%	3.9%	4.4%	4.6%	2.6%	1.9%	3.7%	2.6%

Solution:

In this case, it is only the objective that changes. The constraints remain unaltered.
The new ROI calculations are as follows:
0.049*x1 + 0.023*x2 + 0.024*x3 + 0.039*x4 + 0.044*x5 + 0.046*x6+ 0.026*x7 + 0.019*x8 + 0.037*x9 + 0.026*x10

```
print(obj2)
```
```
[0.049 0.023 0.024 0.039 0.044 0.046 0.026 0.019 0.037 0.026]
```

The model is run in the same way as before in Gurobi, except this time we ask Gurobi to maximize the new objective function:

```
BudgetModel2 = gp.Model() # initialize an empty model

BudgetModX2 = BudgetModel2.addMVar(10) # tell the model how many variables there are
# must define the variables before adding constraints because variables go into the constraints
BudgetModCon2 = BudgetModel2.addMConstrs(A2, BudgetModX2, sense, b2) # add the constraints to the model
BudgetModel2.setMObjective(None,obj2,0,sense=gp.GRB.MAXIMIZE) # Ask Gurobi to Maximize the objective

BudgetModel2.Params.OutputFlag = 0
BudgetModel2.Params.TimeLimit = 3600
```

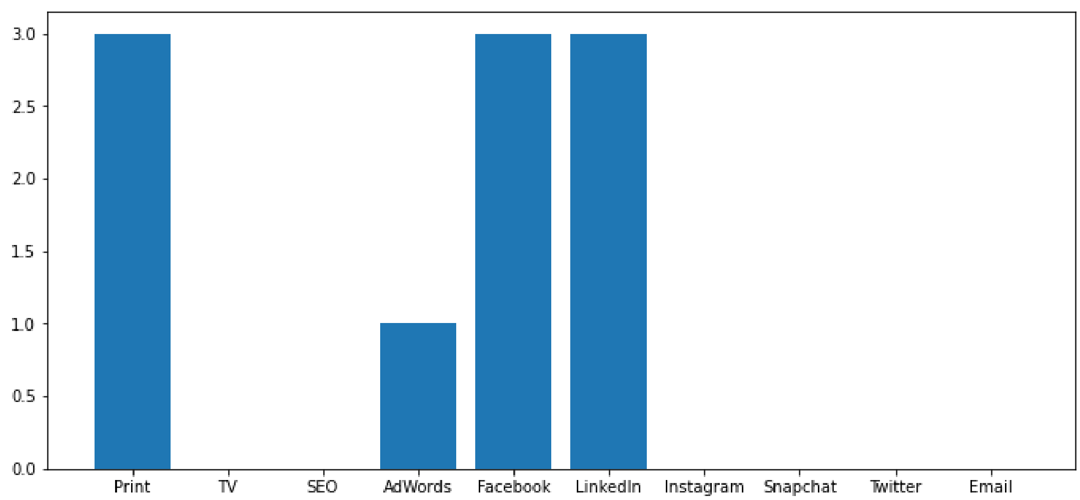
```
Restricted license - for non-production use only - expires 2023-10-25
```

```
[ ] BudgetModel2.optimize() # solve the LP
```

```
[ ] BudgetModel2.objVal # optimal ROI level

0.45600000000000007
```

```
[ ] BudgetModX2.x

array([3., 0., 0., 1., 3., 3., 0., 0., 0., 0.])
```

Results of Optimization:



Takeaways:
This shows that an investment of 10 million will give an ROI of 0.456 million (i.e) 4.56% when the investment is in the proportions of 3 million (print), 0(TV), 0 (Seo), 1 million (AdWorks), 3 million (Facebook), 3 million (LinkedIn), 0 (Instagram) , 0 (Snapchat), 0 (Twitter) and 0 (Email).

Question 5:

Are the allocations the same? Assuming the first ROI data is correct, if you were to use the second allocation (the allocation that assumed the second ROI data was correct) how much lower would the objective be relative to the optimal objective (the one that uses the first ROI data and the first allocation)? Assuming the second ROI data is correct, if you used the first allocation how much lower would the objective be relative to the optimal objective? Do you think the third constraint above, based on your boss' experience, is useful?

Solution:

This is in ways a type of sensitivity analysis to see how the optimal solution works when the platform-wise ROIs are quite different. In our case it leads to a 40%+ reduction in returns in both cases, which goes to show impactful platform ROIs are in deciding the allocations.

```
[56] a = obj1
     c = BudgetModX2.x
     print('second solution in first roi:', a@c)

     print("diff:", BudgetModel1.objVal - a@c)
```

```
second solution in first roi: 0.252
diff: 0.20400000000000007
```

Thus, utilizing the allocation from the second ROI objective in the first ROI objective function leads to total ROI being $0.204M lower, which is a 45% reduction in return

2. Assuming the second ROI data is correct, if you used the first allocation how much lower would the objective be relative to the optimal objective?

```
[57] d = obj2
     e = BudgetModX1.x
     print('first solution in second roi:', d@e)

     print("diff:", BudgetModel2.objVal - d@e)
```

```
first solution in second roi: 0.264
diff: 0.19200000000000006
```

Thus, utilizing the allocation from the first ROI objective in the seocnd ROI objective function leads to total ROI being $0.192M lower, which is a 42% reduction in returns

In order to understand if capping the spends on each of the platforms at 3M is meaningful or not, it would be a good idea to run the model without the constraint. In doing so, you realize that regardless of whether you use the 1st ROI distribution or the second, the budget gets split between 2 platforms and shows an overall improvement in ROI by less than 2%. Putting all your eggs in 1 basket or in this case 2, considerably elevates the risk associated with your portfolio. In other words, the 3M constraint is just a way to curtail risk while diversifying your portfolio and hence, it makes sense to keep it, especially considering how overall ROI growth of 2% is not extremely a worthy tradeoff for the risk it brings with it.

```
BudgetModX5_1.x

array([0., 5., 0., 0., 0., 0., 0., 0., 0., 5.])
```

```
[67] BudgetModX5_2.x

array([5., 0., 0., 0., 5., 0., 0., 0., 0., 0.])
```

Question 6:

To explore this further perform some analysis of how your optimal allocation would change based on changes in the ROI data. Use the first ROI data as your starting point. By how much could each advertising medium's ROI increase or decrease and still result in the same optimal allocation you found in step (3)?

Solution:

In order to understand how much of an ROI change is required to move the needle at an overall returns level, we would first need to understand what level of change wouldn't change overall returns. This can be achieved with the help of sensitivity analysis.

Performing sensitivity analysis on the objective function , we understand the range of ROIs for each platform where the optimal corner remains unchanged.

```python
BudgetModX1.SAObjLow #Range of obj where optimal corner stays the same
```

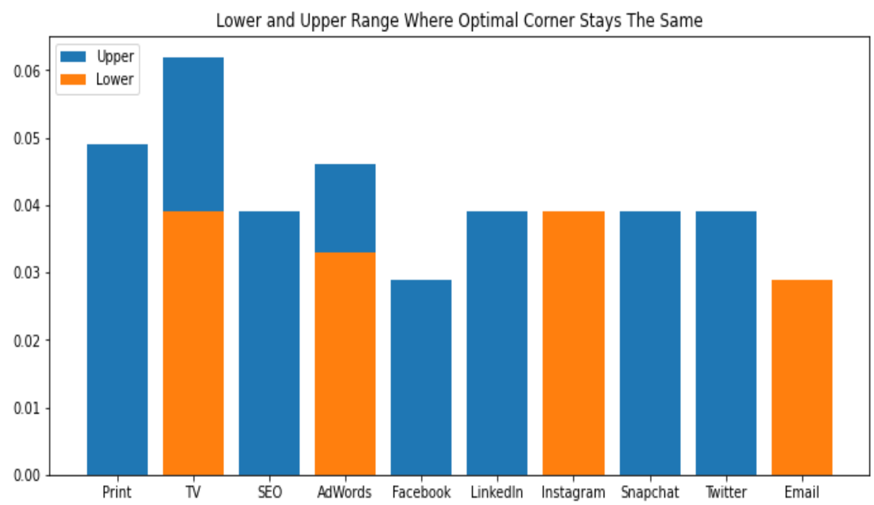
```
array([ -inf, 0.039,  -inf, 0.033,  -inf,  -inf, 0.039,  -inf,  -inf,
        0.029])
```

```python
BudgetModX1.SAObjUp
```

```
array([0.049, 0.062, 0.039, 0.046, 0.029, 0.039,   inf, 0.039, 0.039,
         inf])
```

Illustrated in the table graph below, we can see that TV's ROI can vary anywhere between 3.9% and 6.2% without changing the allocation or optimal corner. A similar notion can be extended to the rest of the channels as well.

	channel	lower	upper
0	Print	-inf	0.049
1	TV	0.039	0.062
2	SEO	-inf	0.039
3	AdWords	0.033	0.046
4	Facebook	-inf	0.029
5	LinkedIn	-inf	0.039
6	Instagram	0.039	inf
7	Snapchat	-inf	0.039
8	Twitter	-inf	0.039
9	Email	0.029	inf



Question 7:

Your boss has gained permission to reinvest half of the return. For example, if the marketing obtains a 4% return in January, the budget of February will be $10M + 10M \times 4\% \times 50\% = \$10.2M$. The monthly ROI for next year is given in Project1.Rdata. The three constraints given by your boss are still in place for each month. What is the optimal allocation for each month?

Solution:

The approach to this would be to look at each month separately and tie together the returns from the previous month to next month's budget. Following are the steps involved in this process:

The modeling part of the part wouldn't be much different from Task 3, however there would be slight differences made to the 'total_budget constraint'. Steps followed are as follows:

1. Adjust the total budget for each month
2. Run a model by changing the total budget constraint to reflect this new number
3. Find the optimal value the model returns. Use this to calculate returns for said month.
   ROI = Returns in said month/ New budget in said month
4. This ROI number can then be used to estimate the budget for the following month
5. Iterate through this process for all months from Jan to December.

The optimal allocations obtained for each month are as follows:

Month	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
January	3.000000	0.000000	0.0	1.333333	0.000000	0.0	2.666667	0.0	0.000000	3.000000
February	3.000000	0.000000	0.0	2.395500	3.000000	0.0	0.000000	0.0	1.791000	0.000000
March	0.000000	0.000000	0.0	3.000000	0.000000	3.0	1.199429	0.0	3.000000	0.000000
April	0.000000	0.000000	0.0	3.000000	0.000000	3.0	3.000000	0.0	1.199707	0.000000
May	1.196177	0.000000	0.0	0.000000	0.000000	0.0	3.000000	0.0	3.000000	3.000000
June	3.000000	0.000000	0.0	0.000000	0.000000	0.0	3.000000	0.0	1.201481	3.000000
July	0.000000	0.000000	0.0	3.000000	1.207644	0.0	3.000000	0.0	3.000000	0.000000
August	2.709695	0.000000	0.0	1.500000	0.000000	0.0	0.000000	0.0	3.000000	3.000000
September	0.607204	0.000000	0.0	3.000000	0.000000	3.0	0.000000	0.0	3.000000	0.607204
October	0.000000	0.000000	0.0	3.000000	0.000000	3.0	3.000000	0.0	0.000000	1.197045
November	3.000000	0.000000	0.0	1.182065	0.000000	0.0	3.000000	0.0	0.000000	3.000000
December	3.000000	2.108393	0.0	0.000000	3.000000	0.0	0.000000	0.0	0.000000	2.108393

Question 8:

A stable budget is defined as a monthly allocation such that for each platform the monthly change in spend is no more than $1M. Is the allocation you found stable? If it isn't, you do not need to solve a new optimization model. Describe how you might model this?

Solution:

Once you have the monthly allocations from task 7 , it's important to estimate the month on month change of platform budget allocation. This will be a good indication of how much the process fluctuates & consequently its volatility.

MoM = Allocation for channel in said month - Allocation for channel in Previous Month

```
# Print MoM values for each month - set to 0 for January by default
diff = [np.zeros(10)]
for j in range(1,12):
    diff.append(opt_allocation[j]-opt_allocation[j-1])

for k in range(len(diff)):
  print(roi_mat.index[k])
  print(diff[k])
  print('\n')
```

```
January
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]


February
[ 0.          0.          0.          1.06216667  3.          0.
 -2.66666667  0.          1.791      -3.        ]


March
[-3.          0.          0.          0.6045     -3.          3.
  1.19942866  0.          1.209       0.        ]


April
[ 0.          0.          0.          0.          0.          0.
  1.80057134  0.         -1.80029329  0.        ]


May
[ 1.19617694  0.          0.         -3.          0.         -3.
  0.          0.          1.80029329  3.        ]
```

Note that we are not looking at percentages here. This MoM number will then serve as commentary on the stability of the allocations.

In this specific scenario, the allocations are highly unstable & fluctuate almost every month. To combat this issue, we would need to redefine the optimization problem.

We could need additional constraints that dictate the Month over Month change in allocation is less than 1M. To be able to do this, we would need 10*12 = 120 decision variables - one for each month-platform combination

A constraint for month over month change would then look as follows:

Jan: a1 to a10 for respect platforms where subscript 1 - print, 2 - TV, 3- SEO and so on with 10 being Email.

Feb: b1 to b10 with similar subscripting

We could define similar variables for **March to December** as well.

A MoM constraint would then look as follows:

Feb MoM: (b1-a1)< 1M, (b2-a2) < 1M .... (b10-a10) < 1M Mar MoM: (c1-b1)< 1M, (c2-b2) < 1M .... (c10-b10) < 1M

We could define MoM constraints for March to December as well.

What would the total number of constraints be then?:

- 12 constraints, 1 for each month, restricting total spend to 10M
- 12 constraints for Print + TV <= Facebook + Email
- 12 constraints for Facebook + LinkedIn + Instagram + Snapchat + Twitter >= 2 * (AdWords +SEO)
- 120 constraints for spend on each channel being less than 3M
- 110 Month over Month constraints
- 120 non negativity constraints

That would be a total of 386 constraints to apply to the new model.