Advertising Project

Group 15

Shreya Bhootda, Ishan Patel, Serena Wu, Shengxiang Wu

Our first task was to determine the allocation of various platforms for a budget given 2 different ROIs. We were also subject to certain constraints such as the amount invested in Print and TV should be no more than the amount spent on Facebook and Email, the total amount used in social media (Facebook, LinkedIn, Instagram, Snapchat, and Twitter) should be at least twice of SEO and AdWords, and for each platform, the amount invested should be no more than $3M. We were also told the total budget cannot exceed $10M.

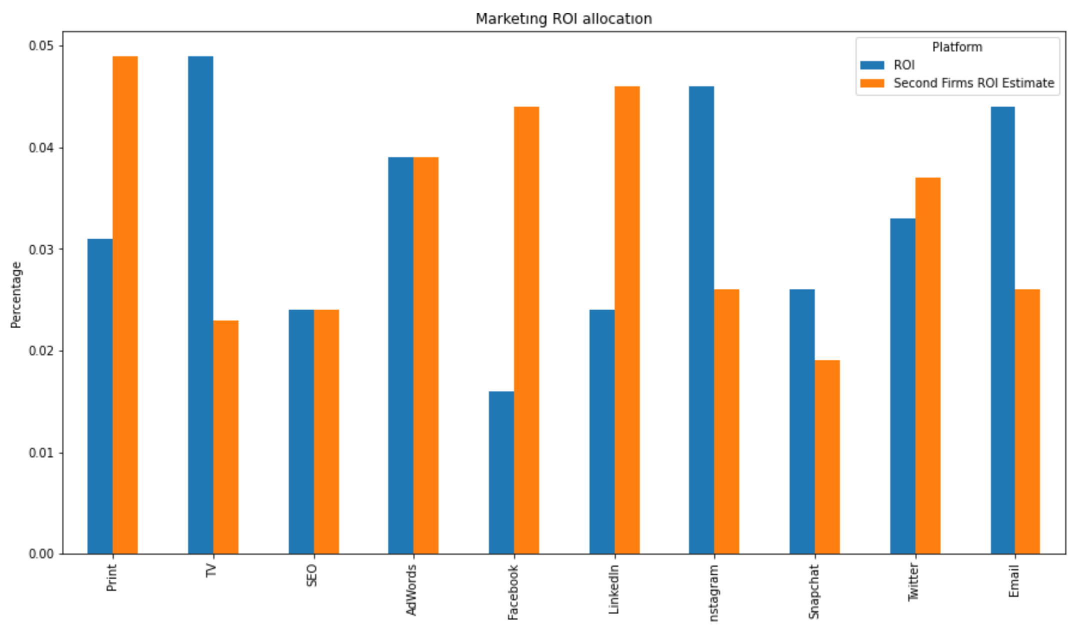

Fig 1.0 : Percentage investment in different advertising mediums

Fig 1.0 illustrates the difference in the marketing strategy between the two firms. We can see that the second firm is concentrating more on advertising via **Print** and social media platforms like **LinkedIn, Facebook and Twitter**. The one thing common among these platforms is that it facilitates putting across the message via text as compared to just visuals i.e pictures, videos etc.

To achieve our goal, we first had to get the ROI data, which was formatted in a .csv file. Using pandas, as shown below, we loaded in this csv file.

```
# read ROI data set
df = pd.read_csv('ROI_data.csv')
```

We were thus able to get the ROIs as shown below.

	Platform	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
0	ROI	0.031	0.049	0.024	0.039	0.016	0.024	0.046	0.026	0.033	0.044
1	Second Firms ROI Estimate	0.049	0.023	0.024	0.039	0.044	0.046	0.026	0.019	0.037	0.026

Next, using the constraints mentioned above and the data from the first row of the "ROI_data.csv" file as the first ROI values and the objective vector, we utilized Gurobi to find an optimal allocation for the platforms. Using the code below, we found the **optimal revenue to be $0.456M with allocations of $3M to TV, Instagram, and Email and $1M to AdWords**.

```
#Instead of hardcoding the ROIs, create a loop to take each ROI from the first row into an objective vector
obj1 = np.zeros(10)
for i in range(10):
    obj1[i] = df.iloc[0, i+1]
```

```
A = np.zeros((13,10))
A[0:10,:] = np.identity(len(obj1)) #Each platform constraint
A[10,:] = [1,1,0,0,-1,0,0,0,0,-1] #No more than FaceBook and Email
A[11,:] = [0,0,2,2,-1,-1,-1,-1,-1,0] #at least twice of AdWords and SEO
A[12,:] = 1

b = np.array([3,3,3,3,3,3,3,3,3,3,0,0,10]) #The RHS of the constraints vector
sense = np.array(['<']*13)
```

```
ROIModel1 = gp.Model() # initialize an empty model

ROIModX1 = ROIModel1.addMVar(10) # tell the model how many variables there are
# must define the variables before adding constraints because variables go into the constraints
ROIModCon1 = ROIModel1.addMConstrs(A, ROIModX1, sense, b) # add the constraints to the model
ROIModel1.setMObjective(None,obj1,0,sense=gp.GRB.MAXIMIZE) # add the objective to the model

ROIModel1.Params.OutputFlag = 0 # tell gurobi to shut up!!
```

```
Restricted license - for non-production use only - expires 2022-01-13
```

```
ROIModel1.optimize() # solve the LP
```

```
ROIopt1 = ROIModel1.objVal # optimal profit level
ROIopt1
```

```
0.45600000000000007
```

```
alloc1 = ROIModX1.x #Solution for allocations
alloc1
```

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

We then did the same approach using the second row from the “ROI_data.csv” file, which contained different ROIs for each platform which the Boss provided, as the objective vector for Gurobi. Based on the **second ROI**, we have found the **optimal allocation of $3M in Print, Facebook and LinkedIn and $1M in AdWords**. This ROI data provided the **same optimal revenue of $0.456M**. However, as our solution shows, we have allocated different amounts of money on different platforms since the ROI provided has been changed while keeping the same optimal revenue.

Our next task was to determine how much the revenues differed from the optimal revenue if the first ROIs were used but with the second allocation, and vice-a-versa. To achieve this, our team did a simple matrix multiplication as shown below. Storing the values from the Gurobi optimization was a huge help as we were able to find both the differences in 2 lines of code.

```
#First ROI but Second Allocation
optRev = np.matmul(obj1,alloc2) #get the revenue based on the ROI and allocation
ROIopt1 - optRev #find the difference between the optimal and calculated revenue
```

```
#Second ROI but First Allocation
optRev = np.matmul(obj2,alloc1)#get the revenue based on the ROI and allocation
ROIopt2 - optRev #find the difference between the optimal and calculated revenue
```

As discussed before, the optimal revenues are the same for each ROI, but the allocations are different. **So if you use the first ROI but second allocation, you stand to lose $0.204M compared to the optimal revenue revenue. If the second ROI is correct but you use the first allocation, you stand to lose $0.192M compared to the optimal revenue value**. Therefore if the boss does choose to mix and match the ROIs and the allocations, it is suggested to use the second ROI with the first allocations as it results in a lower revenue loss.

We also found that the third constraint (the amount invested for each platform should be no more than $3M) can be useful so you don't invest all your money in a small number of platforms, therefore increasing your risk in your portfolio when the ROI forecast is inaccurate. It allows diversification while keeping risk in check.

We then explored some further analysis of how our optimal allocation would change based on changes in the ROI data. Using sensitivity analysis, we found that the ROI of [Print, TV, SEO, AdWords, Facebook, LinkedIn, Instagram, Snapchat, Twitter, Email] need to decrease to [-inf, 0.039, -inf, 0.033, -inf, -inf, 0.039, -inf, -inf, 0.029] in order to change our optimal solution. On the other hand, [Print, TV, SEO, AdWords, Facebook, LinkedIn, Instagram, Snapchat, Twitter, Email] will need to increase to [0.049, 0.062, 0.039, 0.046, 0.029, 0.039, inf, 0.039, 0.039, inf] to change our optimal solution. This was achieved using the code snippets below.

```
ROIModX1.SAObjLow #Smallest objective value at which the current optimal basis would remain optimal
```

```
ROIModX1.SAObjUp #Largest objective value at which the current optimal basis would remain optimal
```

We were then given new information that our boss has gained permission to reinvest half of the return. We were thus tasked to determine the optimal allocation for each month in the year given this new constraint while keeping in mind the previous 3 constraints. To achieve this, **we assume that every month we get 10 millions to invest into these 10 platforms, and half of the return each month can be used as the budget for next month**. We are using the first ROI as our forecasting ROI for each month. After we find out what our optimal allocation is for a specific month, we use the real ROI of that month to calculate the actual return. Half of the actual return will be added to the 10 million budget next month, so the total budget for each month will not be the same.

The results below show the optimal allocation, budget, and return of money with plots. We observe that the allocation of budget for all channels except Adwords is the same. TV, Instagram, and Email receive 3 millions every month; Adwords receives between 1 and 1.2 million every month; Other channels receive 0. The peak in returns is in July (0.403674 millions), while the month with the minimum of returns is March (0.317609 millions). We also observe that the budget in April and October are the lowest of all months. This is because March and September had particularly low returns, which will affect their next month's budget, since each month's budget consists of 10 millions plus half of the returns from last month. On the other hand, since July and November had high returns, their next months (August and December) had higher budgets compared to the rest of the months.

	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email	Return	Budget
January	0.0	3.0	0.0	1.000000	0.0	0.0	3.0	0.0	0.0	3.0	0.360000	10.000000
February	0.0	3.0	0.0	1.180000	0.0	0.0	3.0	0.0	0.0	3.0	0.347840	10.180000
March	0.0	3.0	0.0	1.173920	0.0	0.0	3.0	0.0	0.0	3.0	0.317609	10.173920
April	0.0	3.0	0.0	1.158804	0.0	0.0	3.0	0.0	0.0	3.0	0.341987	10.158804
May	0.0	3.0	0.0	1.170994	0.0	0.0	3.0	0.0	0.0	3.0	0.369814	10.170994
June	0.0	3.0	0.0	1.184907	0.0	0.0	3.0	0.0	0.0	3.0	0.394287	10.184907
July	0.0	3.0	0.0	1.197143	0.0	0.0	3.0	0.0	0.0	3.0	0.403674	10.197143
August	0.0	3.0	0.0	1.201837	0.0	0.0	3.0	0.0	0.0	3.0	0.386477	10.201837
September	0.0	3.0	0.0	1.193239	0.0	0.0	3.0	0.0	0.0	3.0	0.320116	10.193239
October	0.0	3.0	0.0	1.160058	0.0	0.0	3.0	0.0	0.0	3.0	0.335363	10.160058
November	0.0	3.0	0.0	1.167681	0.0	0.0	3.0	0.0	0.0	3.0	0.395875	10.167681
December	0.0	3.0	0.0	1.197937	0.0	0.0	3.0	0.0	0.0	3.0	0.353324	10.197937

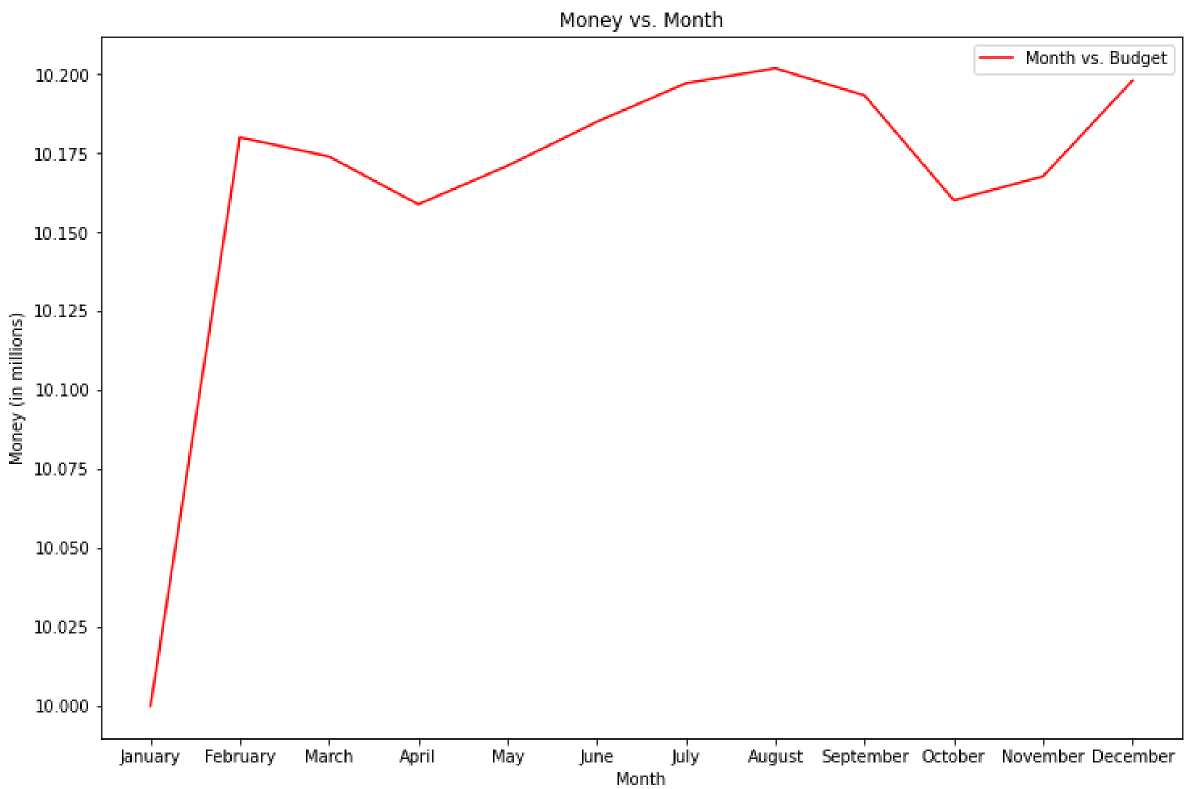

Fig 2.0 : Budget across months

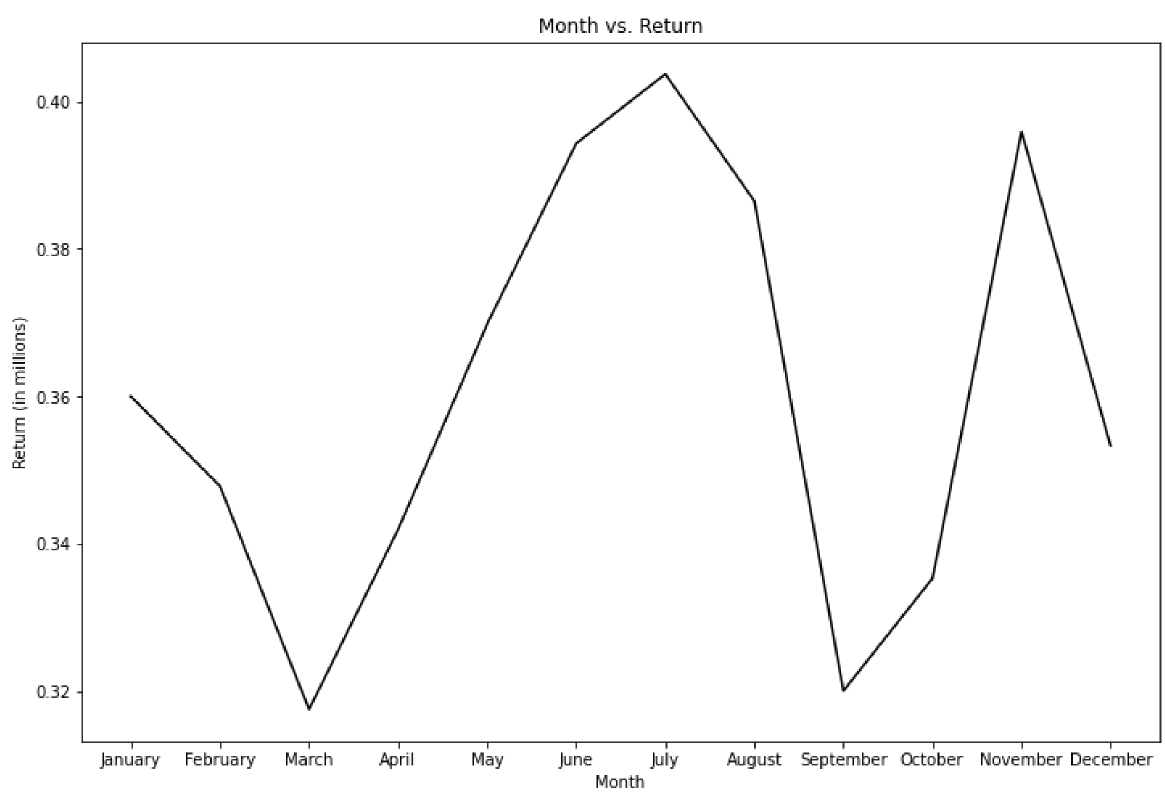

Fig 3.0 : Change in budget with months

As observed from Fig 2.0 and 3.0, we can see that our budget and total amount of money are increasing steadily, but the return we earned changes a lot. The return in October is considerably low although the budget of October is over 12.2 millions.

Based on our result, we can say that for every month, the money we gain as a return is not guaranteed due to the changing of the real ROI rate. However, such a process is stable because our allocation is based on our forecasting ROI data, which does not change, and our allocation for each platform never increased dramatically (by more than $1M) from the previous month.

Therefore, to keep it stable, we should keep using the same forecasting ROI for every month to optimize our allocation instead of changing into the real ROI every month. In addition, we could have a constraint that limits the max amount of allocation on any platform, for example, we can say that the allocation on any platform can be at most one million than the allocation of any other platform.