Marketing Budget Allocations

Fall 2023

Prepared by Company X's Marketing Team

Authors:
Victoria Liu, Vaishnavi Ganesh, Sanjana Nayak,
Sameer Ahmed, Teagan Milford

Optimizing Company X's Marketing Allocation Budget

The objective of this report is to suggest an optimized strategy for budget allocation to different marketing mediums based on predicted return on investment (RoI). The team of data scientists used linear programming to build a simple marketing budget allocation strategy.

Initial and secondary estimates of RoI of each marketing medium are available in the following table and in the CSV file 'ROI_data'.

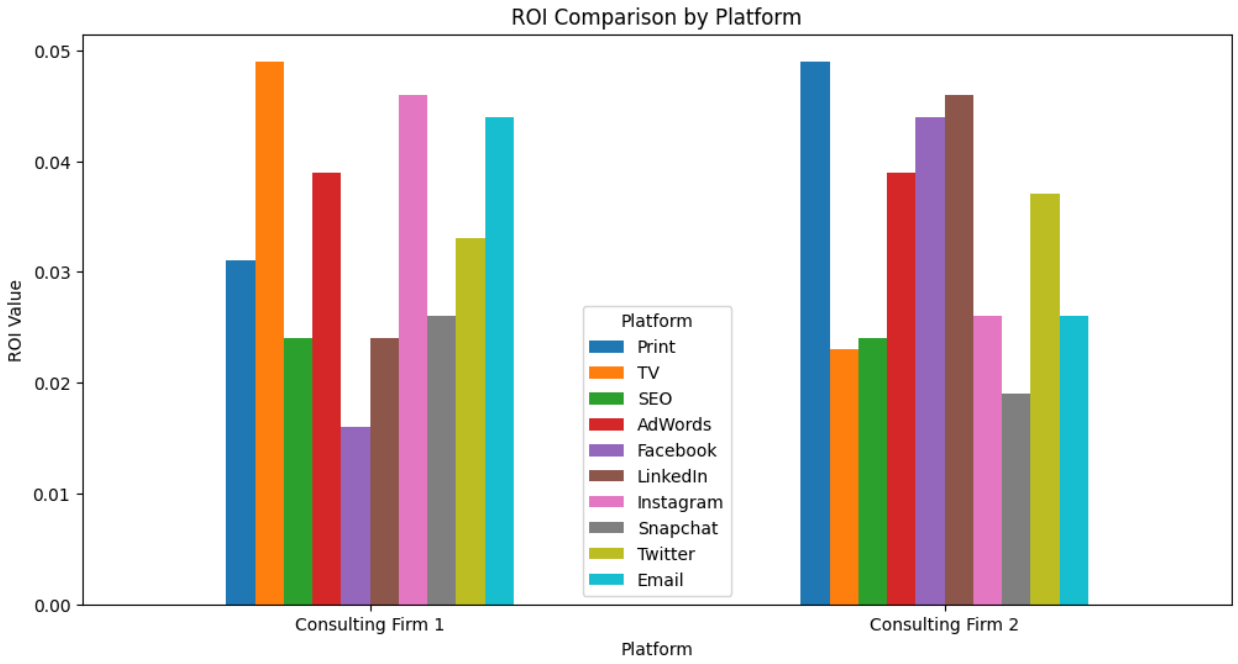
Platform	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
ROI	3.1%	4.9%	2.4%	3.9%	1.6%	2.4%	4.6%	2.6%	3.3%	4.4%

Estimates of RoI from Consulting Company 1

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%

Estimates of RoI from Consulting Company 2

The following chart compares the RoI values from each company, stratified by media platform, conveying the variety present in the predictions.



The overall marketing budget for our Company X is $10M. Additional constraints to the budget include the following:

- 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 the amount used for ‘SEO’ and ‘AdWords’;
- For each platform, the amount invested should be no more than $3M.

The objective function and constraints are considered in the linear program method using the following code:

```
# initializing investment to be 10M dollars
investment=10

# accessing the first row of data from the csv file, RoI data from consulting
firm 1
i=0

# Assuming the first row contains the ROI data and the second row is for the
Second Firm's ROI Estimate
roi_data1 = C1.iloc[i]
roi_data1 = roi_data1[1:]

# initializing a Gurobi model
adMod1 = gp.Model()
# creating a dictionary to hold decision variables
adModX1 = {}

# creating decision variables and adding them to the dictionary dynamically
for platform, roi_percentage in roi_data1.items():
    variable_name = f"{platform}_Var"
    adModX1[platform] = adMod1.addVar(vtype=gp.GRB.CONTINUOUS,
name=variable_name)

# setting the objective function
adMod1.setObjective(gp.quicksum(roi_data1[platform] * adModX1[platform] for
platform in roi_data1.index), sense=gp.GRB.MAXIMIZE)
```

```
# initializing the number of constraints
conlist1=[0]*3

# Budget constraint: Total budget allocated for each channel <= 10M
conlist1[0] = adMod1.addConstr(gp.quicksum(adModX1[platform] for platform in
roi_data1.index) <= investment)

# print + TV <= Facebook+Email -> (Facebook+Email) - (Print+TV) >= 0
conlist1[1] = adMod1.addConstr(adModX1['Facebook'] + adModX1['Email'] >=
adModX1['Print'] + adModX1['TV'])
```

```
conlist1[2] = adMod1.addConstr(adModX1['Facebook'] + adModX1['LinkedIn'] +
adModX1['Instagram'] + adModX1['Snapchat'] + adModX1['Twitter'] >= 2 *
(adModX1['SEO'] + adModX1['AdWords']))

# Social media >= 2(SEO+AdWords) -> Social media - 2(SEO+AdWords) >= 0
for platform, variable in adModX1.items():
    conlist1.append(adMod1.addConstr(variable <= 3))
```

Linear Programming: Estimated RoI from Consulting Company 1

The following code runs linear programming using the estimated RoI from Consulting Company 1. It considers all three constraints, as well as the $10M overall budget, when optimizing. The output of this block is the optimal maximum RoI and the optimal allocation of the budget to the marketing mediums.

```
# Running the model using these constraints and printing the final RoI and the
allocation for each channel
adMod1.Params.OutputFlag = 0
adMod1.optimize()

print("Optimal maximum RoI:",round(adMod1.objVal,3))
print("Allocation of budget on channels:",adMod1.x)
```

```
Optimal maximum RoI: 0.45600000000000007
Allocation of budget on channels:
[0.0, 3.0, 0.0, 1.0, 0.0, 0.0, 3.0, 0.0, 0.0, 3.0]
```

This model tells us that the maximum RoI will be **$456,000**, given that we invest money in the following channels:

Medium	Allocation
Print	$0
TV	$3,000,000
SEO	$0
AdWords	$1,000,000
Facebook	$0
LinkedIn	$0
Instagram	$3,000,000

Snapchat	$0
Twitter	$0
Email	$3,000,000

Linear Programming: Estimated RoI from Consulting Company 2
The following code runs linear programming using the predicted RoI from Consulting Company 2. It also considers the overall $10M budget and all three constraints. The output is the optimal maximum RoI and the optimal budget allocation.

Note: the code is similar to the above code, although variables refer to Consulting Company 2 rather than 1 (*Example: adMod1 becomes adMod2*).

```
# Running the model using these constraints and printing the final RoI and the
allocation for each channel
adMod2.Params.OutputFlag = 0
adMod2.optimize()

print("Optimal maximum RoI:",adMod2.objVal)
print("Allocation of budget on channels:",adMod2.x)
```

```
Optimal maximum RoI: 0.45600000000000007
Allocation of budget on channels:
[3.0, 0.0, 0.0, 1.0, 3.0, 3.0, 0.0, 0.0, 0.0, 0.0]
```

This model tells us that the maximum RoI will be **$456,000**, given that we invest money in the following channels:

Medium	Allocation
Print	$3,000,000
TV	$0
SEO	$0
AdWords	$1,000,000
Facebook	$3,000,000
LinkedIn	$3,000,000
Instagram	$0

Snapchat	$0
Twitter	$0
Email	$0

Initial Analysis

While the optimal budgetary allocations from the two companies are different, the overall RoI is the same. With Company 1's RoI numbers, the model suggests allocating the budget between TV, Adwords, Instagram, and Email. Running Company 2's RoI numbers yields the suggestion of allocating the budget between Print, AdWords, Facebook, and LinkedIn. Neither model suggests allocating funding to SEO, Snapchat, or Twitter, based on the initial RoIs.

Medium	Allocation based of Consulting Company 1	Allocation based of Consulting Company 2
Print	$0	$3,000,000
TV	$3,000,000	$0
SEO	$0	$0
AdWords	$1,000,000	$1,000,000
Facebook	$0	$3,000,000
LinkedIn	$0	$3,000,000
Instagram	$3,000,000	$0
Snapchat	$0	$0
Twitter	$0	$0
Email	$3,000,000	$0
Max RoI	**$456,000**	**$456,000**

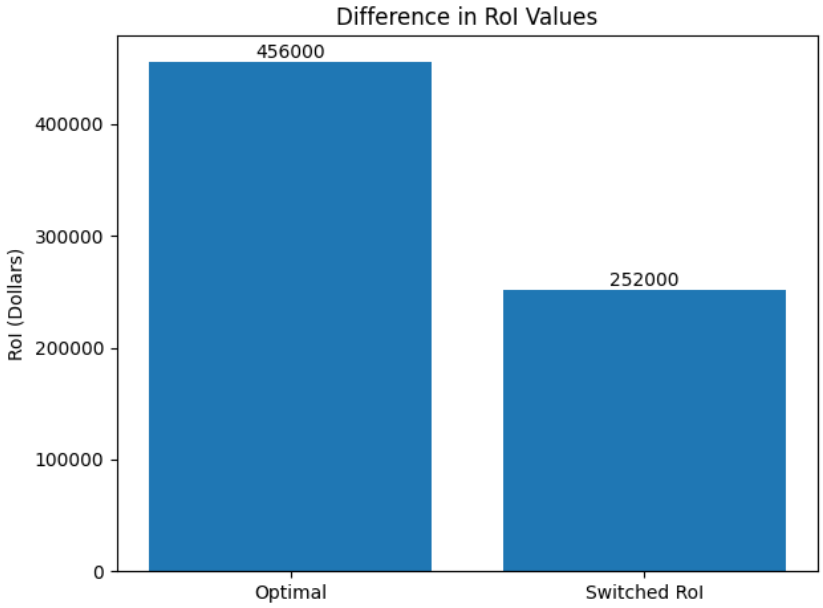
Cross-RoI Analysis

If Company 1's RoI values are correct and the Company 2's allocation suggestion were used, the RoI would be **$204,000 less than the optimal objective**.

```
# Getting the model from the solution got from RoI data from consulting firm 2
alloc2 = adMod2.getAttr("X")

# Allocating the money from first optimal solution into second RoI based model
```

```
roi1_alloc2 = sum(roi_data1[i] * alloc2[i] for i, coefficient in
enumerate(roi_data1.tolist()))

print("RoI after allocating results of second allocation into the optimal
solution:",roi1_alloc2)
print("Difference between optimal RoI and the second allocation based
RoI:",round(adMod2.objVal-roi1_alloc2,3))
```

```
RoI after allocating results of second allocation into the optimal
solution: 0.252
Difference between optimal RoI and the second allocation based RoI: 0.204
```

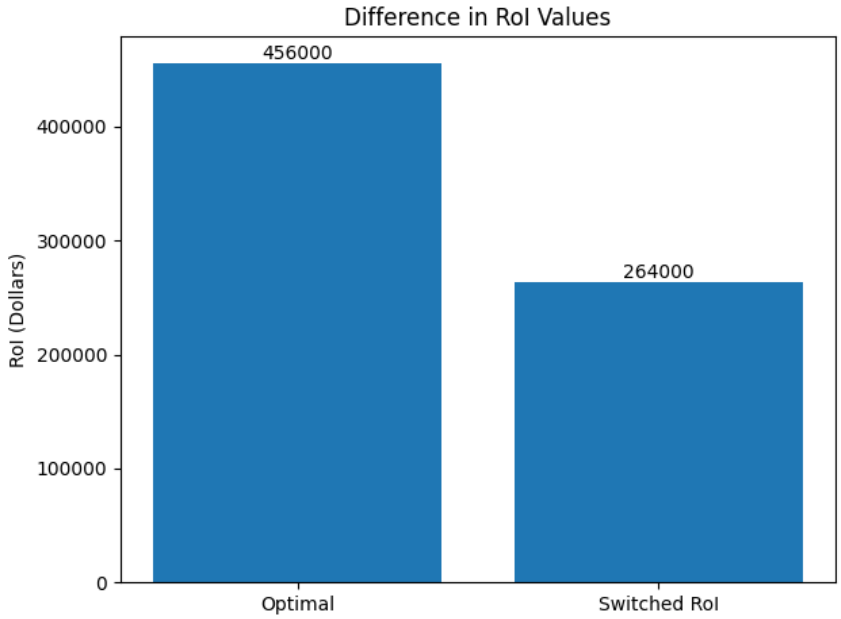


Conversely, if Company 2's RoI values are correct yet Company 1's numbers were applied, the RoI would be **$192,000 less than the optimal objective**.

```
# Getting the model from the solution got from RoI data from consulting firm 1
alloc1 = adMod1.getAttr("X")

# Allocating the money from second optimal solution into first RoI based model
roi2_alloc1 = sum(roi_data2[i] * alloc1[i] for i, coefficient in
enumerate(roi_data2.tolist()))

print("RoI after allocating results of second allocation into the optimal
solution:",roi2_alloc1)
print("Difference between optimal RoI and the second allocation based
```

```
RoI:",round(adMod1.objVal-roi2_alloc1,3))
```

```
RoI after allocating results of second allocation into the optimal
solution: 0.264
Difference between optimal RoI and the second allocation based RoI: 0.192
```



This analysis highlights the significance of RoI predictions provided by the consulting entities. Variance in RoI predictions cause differences in allocation strategies which has huge impacts in our return value. Because of this, it is vital that we move forward with the allocation strategy for the consulting company we trust more.

If we cannot tell which company is more reliable, we should choose the first strategy. This is because, even if it is incorrect, it will lead to the highest RoI.

Evaluating the Constraints

Considering the information that these scenarios yield, the third constraint of allocating only up to $3M per marketing medium is not useful. The following model proves that lifting this constraint increases the amount of RoI that can be obtained from optimal allocation, which amounts to **$465,000**. This is $9000 higher than the RoI attained from both sets of values, given that the third constraint was active.

```
# setting the objective function
adMod3.setObjective(gp.quicksum(roi_data1[platform] * adModX3[platform] for
platform in roi_data1.index), sense=gp.GRB.MAXIMIZE)
```

```
# initializing the number of constraints
conlist1=[0]*3

# Budget constraint: Total budget allocated for each channel <= 10M
conlist1[0] = adMod3.addConstr(gp.quicksum(adModX3[platform] for platform in
roi_data1.index) <= investment)

# print+Tv <= Facebook+Email -> (Facebook+Email) - (Print+TV) >= 0
conlist1[1] = adMod3.addConstr(adModX3['Facebook'] + adModX3['Email'] >=
adModX3['Print'] + adModX3['TV'])

# Social media >= 2(SEO+AdWords) -> Social media - 2(SEO+AdWords) >= 0
conlist1[2] = adMod3.addConstr(adModX3['Facebook'] + adModX3['LinkedIn'] +
adModX3['Instagram'] + adModX3['Snapchat'] + adModX3['Twitter'] >= 2 *
(adModX3['SEO'] + adModX3['AdWords']))

# Removing the constraints of allocating <= 3M for each channel
# for platform, variable in adModX3.items():
    # conlist1.append(adMod3.addConstr(variable <= 3))

# running the model using these constraints and printing the final RoI and the
allocation for each channel
adMod3.Params.OutputFlag = 0
adMod3.optimize()

print("Optimal maximum RoI:",round(adMod3.objVal,3))
print("Allocation of budget on channels:",adMod3.x)
```

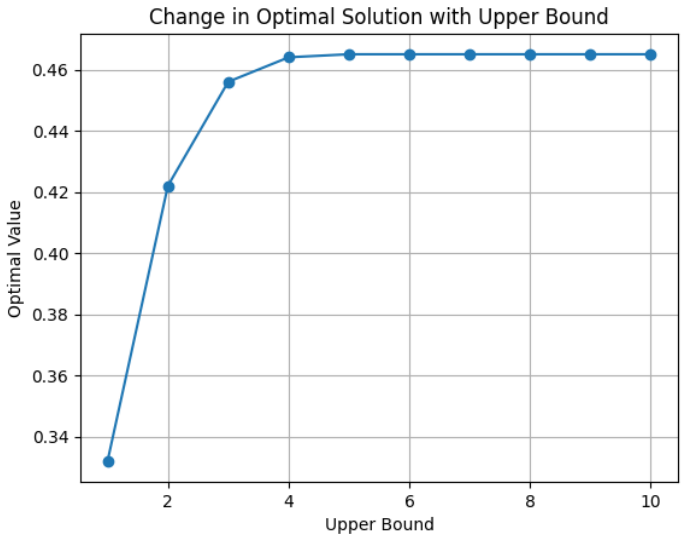
```
Optimal maximum RoI: 0.465
Allocation of budget on channels: [0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 5.0]
```

This model tells us that the RoI will be **$465,000**, given that we invest money in the following channels:

Medium	Allocation
Print	$0
TV	$5,000,000
SEO	$0

AdWords	$0
Facebook	$0
LinkedIn	$0
Instagram	$0
Snapchat	$0
Twitter	$0
Email	$5,000,000

The following graph shows the change in optimal solution with the change of the 3rd constraint. The graph's curve begins to level out when each allocation reaches $4 million, rather than the $3 million mark. This means that the third constraint is not effective.



To explore the importance of the third constraint, further analysis was conducted. The central focus was to question how much each marketing medium's RoI could increase or decrease while still resulting in the same optimal allocation as found with Company 1's numbers. The following code outputs the upper and lower bounds for each medium by using sensitivity analysis.

```
upper_bound=adMod1.SAObjUp
lower_bound=adMod1.SAObjLow
channels=C1.columns[1:]
bounds=pd.DataFrame({'Lower Bounds':lower_bound,'Upper
Bounds':upper_bound},index=channels)
display(bounds)
```

	Lower Bound	Upper Bound
Print	-inf	.049
TV	.039	.062
SEO	-inf	.039
Adworks	.033	.046
Facebook	-inf	.029
Linkedin	-inf	.039
Instagram	.039	inf
Snapchat	-inf	.039
Twitter	-inf	.039
Email	.029	inf

The lower bound and upper bound depict the range within which the RoI can vary while keeping the same optimal solution. This implies that the ROI of Email and Instagram can increase infinitely and this still would not have any impact on the budget allocation. This indicates that constraints 1 and 2 are hindering us from finding the optimal solution. Similarly, the ROI for Print, SEO, Facebook, LinkedIn, Snapchat and Twitter can decrease to negative infinity without having any impact on the budget allocation.

Reinvestment Strategy

After gaining permission to reinvest half of the return from the previous month, the investments, allocations, and returns were recalculated.

	Unnamed: 0	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email
0	January	4.0	3.6	2.4	3.9	3.0	3.5	3.6	2.25	3.5	3.5
1	February	4.0	3.9	2.7	3.8	4.3	3.2	2.7	1.80	3.7	3.5
2	March	3.5	2.9	3.1	3.8	2.4	4.1	3.7	2.60	4.2	2.5
3	April	3.8	3.1	2.4	4.4	2.4	3.8	3.7	2.50	3.6	2.9
4	May	3.5	3.2	1.9	3.4	2.7	2.7	3.9	2.20	4.5	3.9
5	June	4.0	3.2	2.7	3.4	3.4	3.0	4.5	2.10	3.8	4.1
6	July	3.9	3.6	2.0	4.4	3.9	3.7	4.3	1.80	4.0	3.8
7	August	4.2	3.3	2.8	4.2	2.0	3.7	3.6	1.50	4.4	4.3
8	September	4.1	2.8	2.5	4.2	2.9	3.7	2.8	2.50	4.0	3.4
9	October	3.0	3.0	3.1	4.6	3.1	3.3	3.2	2.30	2.5	3.2
10	November	4.8	3.3	2.7	4.1	2.9	3.6	4.2	3.00	3.1	4.1
11	December	4.8	4.0	1.9	3.7	4.2	3.6	2.6	2.90	3.6	3.7

Dataframe with Monthly RoI for Different Mediums

The optimal allocation for the first month was calculated based on the monthly RoI above. Then, half of the RoI from the previous month was added to the budget for the next month (10 million). Thus, as monthly profits improve, future investments tend to grow due to increased access to money for reinvestment funding.

```
# initializing investment to be 10M dollars
investment=10

# initializing lists to store the RoI, monthly investment and allocation for
each channel for all months
roi=[]
investment_monthly=[]
allocation=[]

# looping over the months of the year
for i in range(len(C2)):
    roi_data1 = C2.iloc[i]
    roi_data1 = roi_data1[1:]
    roi_data1 = roi_data1*0.01

    # creating a model for each month
    adMod=gp.Model()

    # creating a dictionary to hold decision variables
```

```
    adModX = {}

    # creating decision variables and adding them to the dictionary dynamically
    for platform, roi_percentage in roi_data1.items():
        variable_name = f"{platform}_Var"
        adModX[platform] = adMod.addVar(vtype=gp.GRB.CONTINUOUS,
name=variable_name)

    # setting the objective function
    adMod.setObjective(gp.quicksum(roi_data1[platform] * adModX[platform] for
platform in roi_data1.index), sense=gp.GRB.MAXIMIZE)
```

```
    # initializing the number of constraints
    conlist1=[0]*3

    # Budget constraint: Total budget allocated for each channel <= 10M
    conlist1[0] = adMod.addConstr(gp.quicksum(adModX[platform] for platform in
roi_data1.index) <= investment)

    # print+Tv <= Facebook+Email -> (Facebook+Email) - (Print+TV) >= 0
    conlist1[1] = adMod.addConstr(adModX['Facebook'] + adModX['Email'] >=
adModX['Print'] + adModX['TV'])

    # Social media >= 2(SEO+AdWords) -> Social media - 2(SEO+AdWords) >= 0
    conlist1[2] = adMod.addConstr(adModX['Facebook'] + adModX['LinkedIn'] +
adModX['Instagram'] + adModX['Snapchat'] + adModX['Twitter'] >= 2 *
(adModX['SEO'] + adModX['AdWords']))

    # Constraints to ensure each variable <= 3
    for platform, variable in adModX.items():
        conlist1.append(adMod.addConstr(variable <= 3))

    # Running the model using these constraints and printing the final RoI and
the allocation for each channel
    adMod.Params.OutputFlag = 0
    adMod.optimize()

    # Appending the RoI, allocation, monthly investments for each month
    roi.append(adMod.objVal)
    allocation.append(adMod.x)
    investment_monthly.append(investment)

    # Re-initializing the investment to 10M
    investment=10
```

```
    # Adding half of the RoI that we received the previous month
    investment=investment+(adMod.objVal*0.5)
```

```
months = C2['Month'].tolist()
columns = C2.columns[1:]
# creating a dataframe with the data we have for each month
roi_d = pd.DataFrame(roi, columns=['RoI Overall'], index=months)
alloc_d = pd.DataFrame(allocation, columns=columns, index=months)
invest_d = pd.DataFrame(investment_monthly, columns=['Monthly Investment'],
index=months)

overall_df = alloc_d.merge(invest_d, left_index=True, right_index=True)
overall_df = overall_df.merge(roi_d, left_index=True, right_index=True)

overall_df = overall_df.reset_index()
display(overall_df)
```

The following allocations reflect the monthly investments and overall RoIs.

	Print	TV	SEO	Adwords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email	Monthly Investment	RoI Overall
January	3.000000	0.000000	0.0	1.333333	0.000000	0.0	2.666667	0.0	0.000000	3.000000	10.000000	0.373000
February	3.000000	0.000000	0.0	2.395500	3.000000	0.0	0.000000	0.0	1.791000	0.000000	10.186500	0.406296
March	0.000000	0.000000	0.0	3.000000	0.000000	3.0	1.203148	0.0	3.000000	0.000000	10.203148	0.407516
April	0.000000	0.000000	0.0	3.000000	0.000000	3.0	3.000000	0.0	1.203758	0.000000	10.203758	0.400335
May	1.200168	0.000000	0.0	0.000000	0.000000	0.0	3.000000	0.0	3.000000	3.000000	10.200168	0.411006
June	3.000000	0.000000	0.0	0.000000	0.000000	0.0	3.000000	0.0	1.205503	3.000000	10.205503	0.423809
July	0.000000	0.000000	0.0	3.000000	1.211905	0.0	3.000000	0.0	3.000000	0.000000	10.211905	0.428264
August	2.714132	0.000000	0.0	1.500000	0.000000	0.0	0.000000	0.0	3.000000	3.000000	10.214132	0.437994
September	0.609498	0.000000	0.0	3.000000	0.000000	3.0	0.000000	0.0	3.000000	0.609498	10.218997	0.402712
October	0.000000	0.000000	0.0	3.000000	0.000000	3.0	3.000000	0.0	0.000000	1.201356	10.201356	0.371443
November	3.000000	0.000000	0.0	1.185722	0.000000	0.0	3.000000	0.0	0.000000	3.000000	10.185722	0.441615
December	3.000000	2.110404	0.0	0.000000	3.000000	0.0	0.000000	0.0	0.000000	2.110404	10.220807	0.432501

Allocations for each Marketing Medium

Is Our Budget Stable?

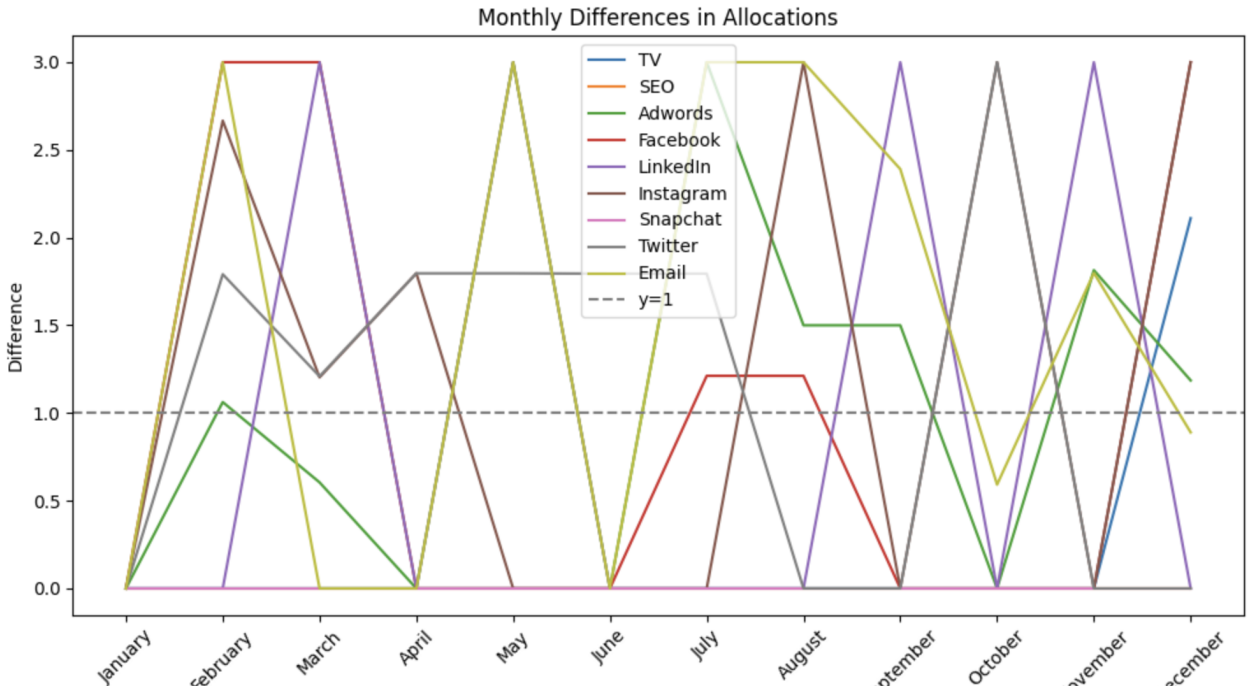
A stable budget is defined as a monthly allocation such that for each platform the monthly change in spend is no more than $1M. To investigate the stability of the monthly proposed budget allocation strategy for each marketing medium, the difference between the allocations for each channel between the months was calculated.

```
df_diff = overall_df.iloc[:, 1:].diff().abs()
```

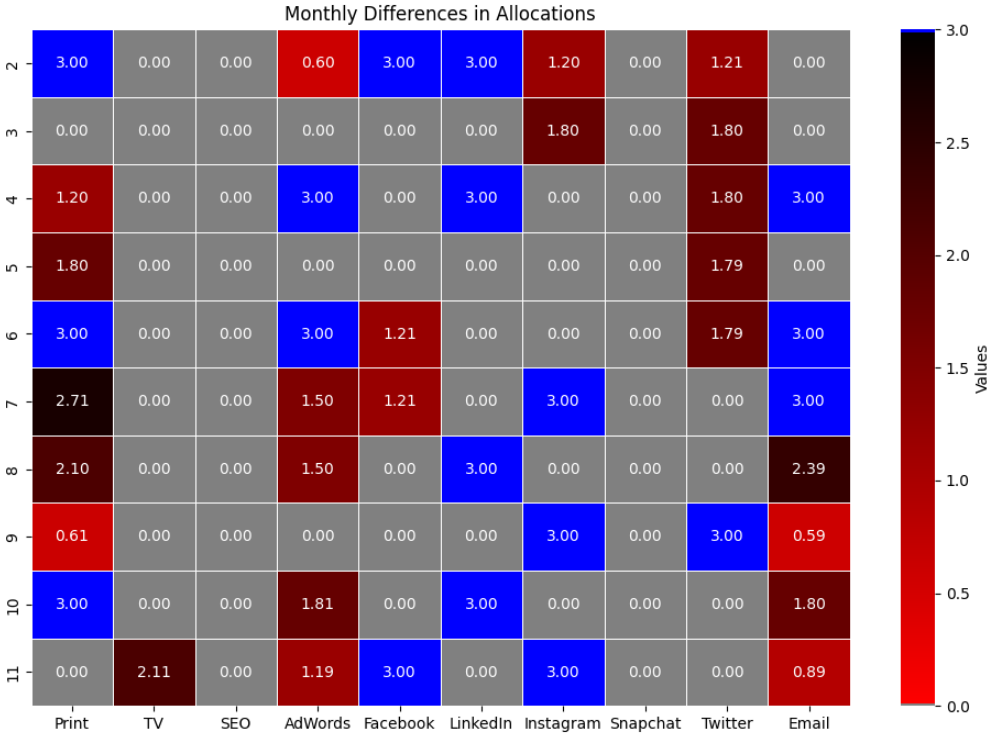
```
#display(df_diff)

# Get the absolute values of the differences in the DataFrame
df_modulus = df_diff.abs()
#display(df_modulus)

# Extract the columns you want to plot (excluding the 'Month' column)
columns_to_plot = df_modulus.columns[:-2]  # Exclude the 'Month' column

# Create a figure and axis for the plot
fig, ax = plt.subplots(figsize=(10, 6))

# Determine the months dynamically
months = overall_df['index'].tolist()

# Loop through the columns and plot each one, aligning data with months
for column in columns_to_plot:
    ax.plot(months, df_modulus[column].tolist(), label=column)  # Exclude the
first month as there's no previous data
```

```
# Add a dotted line at y=1
ax.axhline(y=1, color='gray', linestyle='--', label='y=1')

# Set labels and title
ax.set_xlabel('Month')
ax.set_ylabel('Difference')
ax.set_title('Monthly Differences in Allocations')
ax.legend()

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```

As a result, the following line chart shows each channel's monthly differences in allocations. The following graph illustrates that **none of the allocation differences were below 1M throughout the year.** This shows that none of the budget allocations were stable. We can find a new model to stabilize the allocations.



Additionally, this heat map shows the difference in allocations from month to month. Blue indicates a difference of greater than $1 million, and red indicates a difference of less than $1 million. The pattern of the heat map illustrates instability in the allocation budget, suggesting that the models are insufficient without alterations.



Introducing an additional constraint that limits the change in allocation to be $1M or less for each channel will force stability. With such a new model, we can find an allocation that satisfies the stability requirement while optimizing for the objective.

Final Takeaways

The analysis conducted here reflects the suggestions made based on limited market information. To maximize returns on investment, it is recommended that the constraint of allocating only up to $3 million per marketing medium is omitted. This yields the most promising returns, based on the information at hand. Additional analysis may be beneficial to predict long-term marketing strategies or to target specific demographic segments of Company X's customer base.