# Problem Statement

The equations which govern delivery outcome and growth in customer base are provided.

The delivery outcome is expressed as below where mi is the money invested in improving the delivery outcome on day i (in millions INR).

Text

Description automatically generated with medium confidence

Similarly, growth in customer base is expressed with the below equation where p refers to the delivery outcome probability, Mi is the marketing spend (in Millions INR) on day I and M\_0 is the marketing spend on day zero. Ci refers to the size of the customer base on day i.

Table

Description automatically generated with low confidence

Using the above relationship and the dataset outlined in Datasets, we need to create a strategy for customer base growth for a 30 day period. We start with

1. Initial marketing spend (M\_0) of 5 Million INR
2. Initial Customer Base size (C\_0) of 1,020,000
3. Constant profit margin of 20%
4. Average order value of 300 INR

Simply put, as we control the variables m\_i and M\_i each day\_i, how can we optimize for overall growth in customer base.

The sample dataset provided is available at [this link](https://drive.google.com/file/d/1aYT4OGtv0jjCeQSQrW3cCkMv6UvVrhzF/view). It contains the columns user\_id & order\_frequency\_per\_week, which can be used to estimate daily orders.

# Approaches

The general framework for solving the problem can be divided into the following steps:

1. Getting an estimate of daily orders using simple pro-rata calculations from the weekly data provided
2. Formulating the problem as an optimization or reinforcement learning process and providing the initial conditions
3. Designing the objective function or the reward function and the constraints
4. Setting up functions to calculate starting capital available, expenditure, order volume, profit, ending capital available for each of the 30 days
5. Running a suitable optimizer (or trying multiple optimizers and seeing which one works best) and training the RL model
6. Collecting the results and exporting the investment schedule as a csv

## Optimization Approach

### Leveraging Solver in Excel

I tried setting the problem up in an excel spreadsheet to get a feel for the complexity of the problem. Excel can only handle 200 decision variables and it was okay for this problem as we needed to find the values of m\_i (money spent on improving the delivery outcome) and M\_i (marketing spend) for 30 days implying there were 60 decision variables.

Since the problem at hand needs daily orders to estimate the profit, I used the weekly order frequency data and aggregated it to find the general proportions of various orders frequencies.

|  |  |  |  |
| --- | --- | --- | --- |
| **order\_frequency\_per\_week** | **order\_frequency\_per\_day** | **num\_users** | **weights** |
| 0 | 0.0000 | 14,595 | 51% |
| 1 | 0.1429 | 3,081 | 11% |
| 2 | 0.2857 | 2,925 | 10% |
| 3 | 0.4286 | 2,425 | 8% |
| 4 | 0.5714 | 1,954 | 7% |
| 5 | 0.7143 | 1,434 | 5% |
| 6 | 0.8571 | 1,009 | 4% |
| 7 | 1.0000 | 640 | 2% |
| 8 | 1.1429 | 341 | 1% |
| 9 | 1.2857 | 182 | 1% |
| 10 | 1.4286 | 109 | 0% |
| 11 | 1.5714 | 33 | 0% |
| 12 | 1.7143 | 18 | 0% |
| 13 | 1.8571 | 11 | 0% |
| 14 | 2.0000 | 1 | 0% |
| 16 | 2.2857 | 1 | 0% |

Given the 28,759 customers with the above daily order frequencies and weights, we can pro-rate any other population sizes to arrive at a daily number of orders for that day. This is useful as we expect to grow the customer base over time and would need a good daily estimate of orders to generate profit and operational cashflows.

The objective function was the ratio of final customer base to the initial customer base. Used the GRG Non-Linear Solver to solve to optimization problem.

Constraints:

1. Trivial constraints were added for the decision variables (spends) to be greater than zero
2. Sensible upper bounds were chosen to be 10M for each of m\_i and M\_i.
3. Constraint were added to avoid situations were the sum of m\_i and M\_i for a particular day exceeded the available capital at the start of the day\_i.

Tried solving two different formulations of the problem in Excel:

1. Adding profit back to next day’s capital available
2. Without adding the profit to next day’s available capital

Both approaches have been attached in the document: **asgnmnt.xlsx** and the results are documented in the Summary section of this document.

Submissions are available as:

1. excel-optimization-15M-only-results.csv
2. excel-optimization-profit-reinvested-results.csv

### Leveraging Scipy’s Minimize Module

Converted the excel spreadsheet (reinvesting profit as capital available only) into a python script to optimize the spends using scipy’s optimize module. The objective function was again chosen as ratio of final customer base to the initial customer base. Given we are dealing with a non-linear optimization problem with constraints and bounds, we could go with SLSQP or trust-constr methods in scipy.optimize.minimize. Chose the trust-constr optimizer because of successful experience with it in the past on other problems.

Similar constraints as outlined in the excel based approach were adopted.

Because this is a local-optimizer and is highly dependent on the starting initialization, it is beneficial to initialize randomly multiple times using something like below and running the optimizer multiple times (100 times).

x0 = np.random.uniform(0.1, 20, 60)

Leverage Kaggle Kernels for experimenting quickly and the notebook [is available here](https://www.kaggle.com/code/deepaksadulla/growth-case-study-scipy-minimize/notebook).

The submission is available as **scipy-optimization-results.csv**

## Reinforcement Learning Approach

Experimented with a reinforcement learning approach, by building a custom open-ai gym environment and leveraging stable-baselines3 package’s Proximal Policy Optimization method to come up with a good investment strategy.

Since this is a reinforcement learning approach, we need to define

1. Current state and observation space
2. Actions – [cost of improving delivery outcome, marketing spend]
3. Timesteps
4. Reward/Penalty structure
5. Environment using the two equations provided, profit and the customer growth calculations given the action
6. How do we incorporate constraint like spend less than available starting capital at a particular timestep?

Learning rate and the number of timesteps are key hyper-parameter that we should decide. Have set number of timesteps arbitrarily to 1M and learning rate was left as default and in another case decreased linearly from 0.001 to 0.0003 as the learning progressed.

The [default learning rate version is available here](https://www.kaggle.com/code/deepaksadulla/growth-case-study-rl-gym/notebook) and the [linear learning rate version is available here](https://www.kaggle.com/code/deepaksadulla/growth-case-study-rl-gym/notebook?scriptVersionId=102751129).

Submissions are available as:

1. rl-optimization-default-learning-rate-results.csv
2. rl-optimization-linear-learning-rate-results.csv

# Summary

All the results from the above approaches are summarized below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Solver** | **model** | **duration** | **device** | **growth** |
| Excel | profit reinvested | NA | CPU | 6.96% |
| Excel | fixed capital (15M) | NA | CPU | -17.97% |
| Scipy optimize | profit reinvested | 2839.9s | CPU | 6.69% |
| RL policy + default learning rate | profit reinvested | 2222.1s | GPU | 5.74% |
| RL policy + linear learning rate | profit reinvested | 2225.2s | GPU | 3.54% |

Notes:

1. Growth column in the above table refers to the ratio of [C\_30 / C\_0 minus 1]
2. Highest customer growth was achieved by the Excel Solver when the profits were reinvested as capital for the next day
3. The objective function for optimization was always C\_30 / C\_0
4. RL’s reward was designed to be [(self.num\_customers / C\_prev) – 1]
5. If we only use the available 15M to drive customer growth, the set of equations and the current model would not allow it. It is not enough capital to drive growth by improving delivery outcomes and marketing spends

# Opportunities for improvement

Some other information that could have been leveraged includes:

1. Order Value fluctuates during the days in a week hence working with actual dates so that we can capture this effect in a 30-day period. Transactions data with date would be more suitable to a get a better estimate and avoid a top-down estimate for daily orders (currently implement daily estimates from weekly level dataset)
2. Multi objective optimization involving profit and customer-base growth would be more suitable from a business standpoint. We will need to scale these different objective manually and then combine them as ∑[(x \* Profit\_i) + ((1-x) \* G\_i / G\_i\_minus\_1)]
3. Optimizers like trust-constr in scipy are prone to finding closer local optimums, using global optimizers will be slower but might be worth it if the business demands even better management of capital
4. Training the reinforcement learning model for more timesteps, trying other methods after carefully understanding the RL literature, tuning the hyperparameters might yield better results given the evolving nature of the problem
5. Leverage IPOPT (or other similar solvers) for non-linear constrained optimization if the number of decision variables increases significantly if one decides to increase the current 30 days’ time window
6. Optimization is faster if we provide the necessary gradients of the objective function and constraints, hence leveraging packages like JAX, pytorch which perform automatic differentiation might be beneficial
7. There could be other formulations for the overall model/objective functions/better constraints/solvers that I cannot think of