VIETDATA.AI
Data - driven • AI - guided

ASSESSMENT TEST

DATA ANALYST

ALGORITHM 2

Wizard Potion

Candidate:

Tan Tai

Dec, 2024

Contents

Problem Statement

In a 3-apprentice potion production business, the goal is to maximize revenue over a defined duration, allowing spell allocation and weekend overtime work.

Each apprentice produces potions daily with potential production enhancements through milestone achievements and the application of spells. However, operational constraints such as limited spell usage, profitability thresholds for weekend work, and varying durations create a complex decision-making environment.

My objectives are:

- **Assess the impact of weekend work:** I will determine the profitability of enabling weekend work based on a predefined revenue thresholds (let's say $800,000).
- **Analyze Duration Scenarios:** I do simulation revenue outcomes for different durations to identify optimal timeframes.

Assumptions

However, there are some assumptions to make the problem clearer.

1. Regular workdays: Monday to Friday (5 days)
2. Each apprentice makes 1 potion/daily initially.
3. Weekend overtime costs $7,000 for all of them, allowing 7 workdays per week.
4. If an apprentice has reached a 10-potion milestone, apply a 7% speed boost to potion-making rate.
5. The spells' effects are temporary and last only for that workday.
6. Spells can be cast on only 2 apprentices per workday, each apprentice can receive 1 spell per day.
7. I can only cast one spell per apprentice and cast spells on 2 apprentices each workday.
8. They are unpaid internships.
9. Initial potion-making rate are equal among the 3 apprentices.

Discharge assumptions

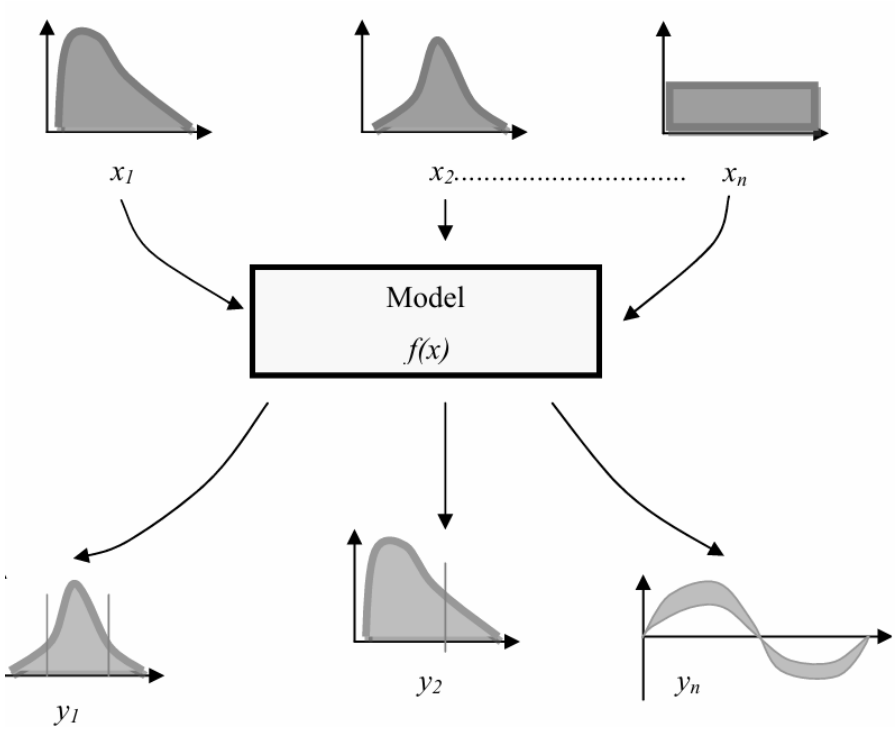
At first hand, I have made some assumptions but they soon were not adopted within the context of my analysis.

- **Casting multiple spells on one apprentice:** Although the problem does not explicitly prohibit casting multiple spells on a single apprentice, I decided to discard this assumption because it does not align with a logical interpretation of this problem.
- **Cumulative spell effects:** I treated all spells as temporary. This approach prevents unrealistic compounding of effects over time.

1 Monte Carlo simulation approach

It is recommended that we should use Monte Carlo to interpret the **randomness** of this kind of problem. Its goal is to determine how **random variation**, **lack of knowledge**, or **error** affects the sensitivity, performance, or reliability of the system that is being modeled. Here, the inputs are randomly generated from probability distributions to simulate the process of producing revenue.

Monte Carlo simulation allows me to model the variability via multiple iterations, providing a robust estimate of all the factors that impact on revenue.



To use Monte Carlo simulation, all we need to do is follow the five simple steps:

- Step 1: Create a parametric model, $Y = f(X_1, X_2, \ldots, X_n)$.
- Step 2: Generate a set of random inputs, $X_{i1}, X_{i2}, \ldots X_{in}$
- Step 3: Evaluate the model and store the results as Y_i .
- Step 4: Repeat steps 2 and 3 for $i = 1$ to n.
- Step 5: Analyze the results using histograms, summary statistics, confidence intervals, etc.

2 Parametric Model

2.1 Variables affect revenue

2.1.1 Spell allocation

The allocation of spells directly affects potion production and revenue. The three spells: Haste, Empower, Lady Luck denoted as S_H, S_E, S_L respectively - they are modeled as binary random variables that determine whether a particular spell is applied to an apprentice on a given day.

Each spell has different effects on potion-making rates and sale prices, which I will demonstrate later.

1. Haste (S_H): **Increases** potion-making rate by +18% on the day it is applied.
2. Empower (S_E): **Increases** potion sale price by \$100 on the day it is applied.
3. Lady Luck (S_L): A 12% chance of doubling potion production for the day.

S_L	1	2
$\mathbb{P}(S_L)$	0.88	0.12

2.1.2 Potion-making rate (R_i)

More potions produced directly increase revenue. The potion-making rate determines how many potions apprentice i produces on day t.

It is influenced by several factors, including the base production rate, milestone boosts, and spell effects.

- **Base rate:** Suppose each apprentice starts at producing 1 potion/day.
- **Milestones speed boost:** A **cumulative** 7% boost to production rate is added after every 10 potions produced.

Let $n_i(t)$ is the number of milestones apprentice i has reached:

$$n_i(t) = \left\lfloor \frac{\text{Cumulative Potions}}{10} \right\rfloor$$

According to spells, $S_H(t) = \begin{cases} 1 & \text{, Apply haste} \\ 0 & \text{, otherwise} \end{cases}$ and $S_L(t) = \begin{cases} 1 & \text{, } p = 0.88 \\ 0 & \text{, } p = 0.12 \end{cases}$
Therefore the formula for potion-making rate of apprentice i on day t is:

$$R_i(t) = [1 + 0.07 \times n_i(t) + 0.18 \times S_H(t)] \times S_L(t)$$

2.1.3 Potion sale price (P_i)

Higher sale prices multiply with the number of potions produced to yield higher daily revenue. It is influenced by the base price and the application of the Empower spell:

- **Base Price:** \$1,000 per potion.
- **Empower Spell**: Increases potion price by \$100 **for that work day**.

Hence, with $S_E(t) = \begin{cases} 1 & \text{, Apply Empower} \\ 0 & \text{, otherwise} \end{cases}$ the formula for sale price per potion is:

$$P_i(t) = 1,000 + 100 \times S_E(t)$$

2.1.4 Weekend work decision (W)

- **Cost:** \$7,000 to add 2 extra workdays.
- **Benefits:** Potential increase in potion production and revenue for the week.

First, I need to determine whether day t is weekday or not by the following formula:

$$Daytype(t) = \begin{cases} \text{Weekday} & \text{if } t \mod 7 \in \{0, 1, 2, 3, 4\} \\ \text{Weekend} & \text{if } t \mod 7 \in \{5, 6\} \end{cases}$$

Moreover, my strategy is enable weekend work overtime at the $800,000 revenue threshold. By the time revenue exceeds this threshold, the potion-making rate is significantly higher because of milestone. These higher potion-making rates maximize the revenue generated on weekends, making the additional cost of $7,000 per weekend more justifiable.

Therefore, the weekend decision is model as $W(t) = 1$ if t is Weekend and $\sum_{k=1}^{t-1}(\text{Daily\_Revenue(k)}) \geq 800,000$. The case $W(t) = 0$ is otherwise.

2.2 Parametric model

Weekday Revenue $R_1(t)$: For $t \in$ Weekdays, $R_1(t) = \sum_{i=1}^{3} R_i(t) \times P_i(t)$

Weekend Revenue $R_2(t)$: For $t \in$ Weekends, $R_2(t) = W(t) \times \left[\sum_{i=1}^{3} R_i(t) \times P_i(t) - 7,000\right]$

Therefore, the total revenue, denote as R can now be expressed as:

$$R = \sum_{t \in \text{Weekdays}^T} \left[\sum_{i=1}^{3} R_i(t) \times P_i(t)\right] + \sum_{t \in \text{Weekends}^T} W(t) \times \left[\sum_{i=1}^{3} R_i(t) \times P_i(t) - 7,000\right]$$

Duration T: The term T in the formula that calculates total revenue represent the duration of this operation. It defines the total number of days over which the revenue is calculated, such as:

- $T = 90 \rightarrow$ 3-month duration
- $T = 180 \rightarrow$ 6-month duration

3 Do the simulation

Now that the parametric model is well-defined, I can proceed to the simulation phase.

3.1 Define Simulation Parameters

- Number of Iterations: Let the Monte Carlo simulation runs 1,000 times.
- Duration of the operation: First, I let $T = 90$
- Threshold for Weekend Work: Set a $800,000 revenue threshold.

3.2 Generate random inputs

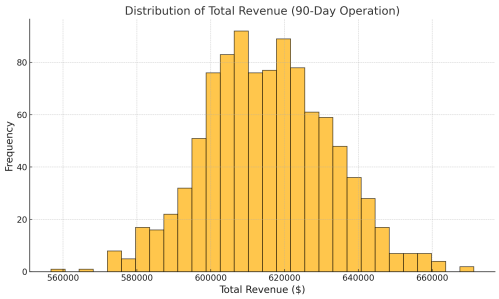
You can see the full Python code in my Github page. Here's the first 5 rows:

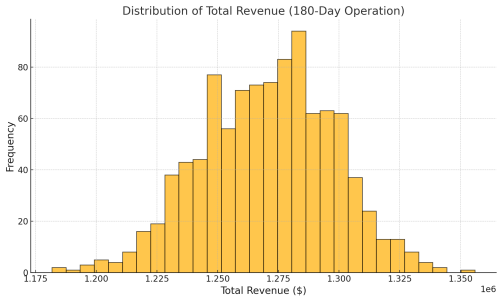
	Simulation	Day	Apprentice	Lady_Luck	Haste	Empower
0	1	1	1	1	0	0
1	1	1	2	1	0	1
2	1	1	3	1	1	0
3	1	2	1	1	0	0
4	1	2	2	2	1	0

3.3 Calculate Potion-Making Rate and Sale Price

	Simulation	Day	Apprentice	Potion_Rate	Sale_Price	Cumulative_Potions
0	1	1	1	1.00	1000.0	1.00
1	1	1	2	1.00	1100.0	1.00
2	1	1	3	1.18	1000.0	1.18
3	1	2	1	1.00	1000.0	2.00
4	1	2	2	2.36	1000.0	3.36
5	1	2	3	1.00	1100.0	2.18
6	1	3	1	1.00	1000.0	3.00
7	1	3	2	1.00	1100.0	4.36
8	1	3	3	1.18	1000.0	3.36
9	1	4	1	1.00	1000.0	4.00

3.4 Distribution Graph





4 Conclusion

1. **Revenue outcomes:**

   - The 90-day operation generated an average total revenue of approximately $615,363, with a maximum of $671,253 and no probability of reaching the $1,000,000 target.
   - The 180-day operation, as I expected, produced higher revenue, with a wider distribution. The average revenue increased substantially, and the probability of reaching $1,000,000 is now non-zero, indicating that longer durations provide more opportunities for milestone boosts and sustained weekend work.

2. **Weekend work efficiency:** With the longer 180-day duration, weekend work was enabled more frequently due to the cumulative revenue threshold ($800,000) being reached earlier and sustained. This, therefore, increased weekend revenue contributions significantly compared to the shorter duration.