EMAIL OPEN TIME OPTIMIZATION USING THOMPSON SAMPLING

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications

by

Shweta Singh
(23419MCA053)

Department of Computer Science
Institute of Science

Banaras Hindu University, Varanasi – 221005

December 2024

CANDIDATE'S DECLARATION

I hereby certify that the work, which is being presented in the report, entitled **Email Open Time Optimization Using Thompson Sampling**, in partial fulfillment of the requirement for the award of the Degree of **Master of Computer Applications** and submitted to the institution is an authentic record of my own work carried out during the period *October-2024* to *November-2024* under the supervision of Prof. S Kathikeyan and Dr. Ankita Vaish. I also cited the reference about the text(s) /figure(s) /table(s) /equation(s) from where they have been taken.

The matter presented in this thesis has not been submitted elsewhere for the award of any other degree or diploma from any Institutions.

Date: Signature of the Candidate

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Date: Signature of the Research Supervisor

The Viva-Voce examination of *Shweta Singh*, M.C.A. Student has been held on ________________.

Signature of
External Examiner

Signature of
Head of the Department

ABSTRACT

In the ever-evolving digital landscape, determining the optimal timing for customer interactions is crucial for enhancing engagement and maximizing revenue. This study leverages Reinforcement Learning (RL), a subset of machine learning, to identify the best times for email openings and other customer interactions. The project employs the Thompson Sampling algorithm, a probabilistic approach that dynamically adapts strategies based on customer behavior and observed outcomes. By balancing exploration (testing new strategies) and exploitation (focusing on proven successful times), the model effectively learns to optimize email open rates. Using simulated data as a controlled environment for learning, the study demonstrates the potential of RL-based techniques to refine decision-making in marketing, paving the way for smarter, data-driven strategies across various communication channels.

Keywords: Reinforcement Learning, Thompson Sampling, Exploration, Exploitation.

TABLE OF CONTENTS

Title Page No.

CHAPTER 6 MODEL ARCHITECTURE

CHAPTER 7 RESULTS AND DISCUSSION

CHAPTER 8 CHALLENGES

CHAPTER 7 CONCLUSION AND FUTURE WORK

CHAPTER 1

INTRODUCTION

1.1 Importance of Timing in Digital Interactions

In today’s fast-paced digital world, customer attention is a valuable and limited resource. The timing of customer interactions, such as sending emails, app notifications, or advertisements, plays a critical role in ensuring these messages are seen and acted upon. Poorly timed interactions risk being ignored, buried in inboxes, or perceived as irrelevant, ultimately leading to lost engagement opportunities and revenue.

To address this, businesses increasingly rely on data-driven approaches to determine the best time to engage customers. Understanding when a customer is most likely to open an email or interact with a notification allows organizations to optimize their strategies, maximizing visibility and effectiveness. However, this requires balancing multiple factors, such as individual preferences, behavioral trends, and varying patterns across different customer segments.

In this project, we leverage **Reinforcement Learning (RL)** to dynamically identify optimal interaction timings, focusing on email open rates. By applying Thompson Sampling algorithm, we explore a robust and adaptive approach to optimize timing strategies, ensuring higher engagement and improved customer satisfaction.

1.2 Objectives Of The Study

The primary objective of this study is to develop a robust and adaptive model that optimizes the timing of email campaigns to maximize customer engagements and open rates. By leveraging **Reinforcement Learning (RL)** and the **Thompson Sampling algorithm**, the study aims to identify the best – performing days for email interactions dynamically.

The specific objectives include:

- Designing a model that learns optimal email timings by balancing **exploration** (testing different days) and **exploitation** (focusing on the best-performing day).
- Evaluating the effectiveness of **Thompson Sampling** in improving decision-making for dynamic environments like email campaigns.
- Analyzing the model's performance through metrics such as **success rates**, **regret**, and **regret percentage**, demonstrating its ability to converge to optimal strategies over time

By achieving these objectives, this study contributes to enhancing customer engagement strategies and showcases the potential of probabilistic approaches in solving real-world marketing problems.

1.3 Overview of Reinforcement Learning (RL)

Reinforcement Learning (RL) is the science of decision-making, where an agent learns to identify the best possible behavior within an environment to achieve maximum rewards. Unlike supervised learning, RL involves the agent interacting with the environment and learning from its responses, similar to how children explore their surroundings and discover the actions that help them accomplish their goals.

In RL, there is no direct supervision. The learner must independently determine the sequence of actions that maximize cumulative rewards. This learning process relies heavily on trial-and-error exploration, where the agent evaluates the effectiveness of its actions based not only on the immediate reward but also on the delayed rewards these actions may yield over time

Key Elements of RL:

- **Agent**: The learner or decision-maker responsible for taking actions.
- **Environment**: The external system or surroundings with which the agent interacts.
- **Policy**: The strategy or rule the agent follows to decide its actions.
- **Reward Signal**: The feedback the agent receives after an action, indication its success or failure.

Applications in Real-World Problems:

- **Robotics**: Learning to perform tasks autonomously.
- **Games**: Developing strategies to defeat opponents or achieve high scores.
- **Autonomous Driving**: Learning to navigate and make decisions in complex traffic environments.

CHAPTER 2

LITERATURE REVIEW

2.1 Background on Reinforcement Learning

Reinforcement Learning (RL) is a branch of machine learning focused on teaching an agent how to make decisions in an environment to maximize cumulative rewards. Unlike supervised learning, RL does not rely on labeled data but instead uses **rewards** and **penalties** as signals to guide the learning process.

The foundations of RL are inspired by **behavioral psychology**, where learning occurs through interaction with the environment. RL incorporates concepts like **trial and error**, where an agent explores different actions and learns which ones lead to better outcomes.

The key feature of RL is its ability to handle **sequential decision-making**. The agent's goal is not just to optimize for immediate rewards but also to maximize future rewards. The requires the agent to learn an **optimal policy**, which is a strategy for selection actions that yield the best long-term benefits.

Over time, RL has evolved with significant contributions, including:

- The development of mathematical frameworks like **Markov Decision Processes (MDPs)** to model environments.
- The introduction of powerful algorithms like **Q-learning**, **SARSA**, and **Thompson Sampling**, which allow agents to learn and adapt in dynamic environments.

2.2 Thompson Sampling and Applications

Thompson Sampling is a probabilistic algorithm used in reinforcement learning for decision-making under uncertainty. It is particularly effective for solving **multi-armed bandit problems**, where the goal is to choose the best action from a set of options to maximize rewards over time.

The algorithm works by maintaining a probability distribution for each option (or action) and updating these distributions based on observed successes and failures. In each trial, Thompson Sampling selects an action by sampling from these distributions, balancing:

- **Exploration**: Trying less-explored options to gather more information.
- **Exploitation**: Choosing the option with the highest estimated probability of success.

This balance helps the model converge to an optimal strategy over time.

Applications of Thompson Sampling:

1. **Online Advertising**: Determining the best ad to display to users based on click-through rates.
2. **Healthcare**: Optimizing treatment strategies by selecting the most effective treatments for patients.
3. **Marketing Campaigns**: Choosing the best times to send promotional emails or notifications to maximize engagement.
4. **Recommendation Systems**: Suggesting content (e.g., movies or products) based on user preferences and feedback.

Background Study:

To explore the relevance of Thompson Sampling in this project, a review of prior work was conducted. The study revealed that:

- Thompson Sampling has been widely applied in marketing and resource allocation, such as choosing optimal ad placements and testing customer preferences through A/B testing.
- Applications in **timing optimization for email campaigns** are limited, making this study a relatively **novel exploration** in the context of marketing strategies.

Novel Application:

While **Thompson Sampling** has been extensively studied for applications like adverting and resource allocation, its **use in timing optimization for email campaigns is relatively less explored** in academic literature, making this study a novel contribution to marketing strategies.

2.3 Relevance of Timing Optimization in Marketing

In marketing, **timing is a critical factor** that determines the effectiveness of customer interactions. The success of campaigns such as email promotions, app notifications, and advertisements heavily depends on reaching customers at the right moment. Poorly timed messages can lead to lower engagement, missed opportunities, and decreased revenue.

By identifying the optimal time to send emails or notifications, businesses can:

- **Maximize Visibility**: Ensure messages are seen by customers before they get buried under other communications.
- **Increase Engagement**: Encourage customers to interact with the content, such as opening emails or clicking links.
- **Enhance Customer Experience**: Deliver communications when customers are most receptive, leading to higher satisfaction.

Current Challenges in Timing Optimization:

1. **Behavioral Variability**: Customer habits and preferences vary significantly, making it difficult to predict the best time to engage.
2. **Data Limitations**: Real-time data on customer behavior is often limited or unavailable.
3. **Dynamic Environments**: Customer preferences may change over time, requiring adaptive strategies.

CHAPTER 3

PROBLEM STATEMENT AND OBJECTIVES

3.1 Problem Defination

Context:
For merchants, **email marketing** is one of the most effective tools for engaging customers and driving revenue. However, the **timing of email campaigns** plays a crucial role in their success. If emails are sent at the right time, they are more likely to be opened, read, and acted upon, leading to higher engagement and conversion rates. On the other hand, sending emails at the wrong time can result in them being overlooked or buried in a crowded inbox, leading to missed opportunities.

Problem:
The challenge arises in determining the **best time** to send emails to maximize their effectiveness. Emails sent at **inappropriate times** risk being ignored, diminishing the potential revenue they could generate. Predicting the optimal days for email campaigns is difficult due to the unpredictability of customer behavior, which varies across different demographics, preferences, and habits.

Moreover, many marketing strategies rely on **traditional or rule-based systems** without leveraging data-driven insights, making it difficult to adapt to changing customer behavior or discover the ideal timing for engagement. This leads to suboptimal decision-making and less effective marketing strategies.

This problem emphasizes the need for an approach that can **dynamically** adjust to customer behavior and provide insights on the best days to send emails, maximizing customer engagement and revenue.

3.2 Objectives of the Project

The primary objectives of this project are as follows:

1. **Model Development Using Reinforcement Learning:**
   The goal is to create a model that uses **Reinforcement Learning (RL)** to determine the optimal day for sending emails. By continuously learning from interactions with the simulated environment (based on customer engagement), the model will identify the best days that maximize email open rates and improve overall customer engagement. This will allow for a dynamic, data-driven approach to email campaign timing, making the campaigns more effective.

2. **Evaluating Thompson Sampling:**
   A key part of the project is to evaluate the effectiveness of **Thompson Sampling**, a popular RL algorithm, in handling the **exploration vs. exploitation** dilemma. The model will explore different days (exploration) to gather data on their effectiveness, while also exploiting the days that are known to yield the highest open rates (exploitation). The objective is to find an optimal balance between these two strategies to enhance email campaign success.

3. **Improvement in Email Campaigns:**
   By utilizing Thompson Sampling, the model aims to improve the overall **performance of email campaigns**. This includes increasing the likelihood that emails are opened, thereby improving customer engagement and maximizing revenue from email marketing efforts.

CHAPTER 4

DATA SIMULATION AND PREPARATION

4.1 Simulated True Probabilities

In this project, **Simulated True Probabilities** represent the likelihood that an email will be opened on each day of the week, from **Sunday to Saturday**. These probabilities are essential for simulating the outcome of each trial (whether an email is opened or ignored) during the model's learning process.

- **Purpose**:
  The true probabilities help **simulate customer behavior**, providing a foundation for the model to explore different days and determine which day maximizes email open rates. These probabilities are used to simulate the **outcome of each trial**: if the randomly generated probability is less than the true probability, the email is **opened** (success), and if it's higher, the email is **ignored** (failure). Although these simulated outcomes guide the learning process, they are **not directly used by the model** for decision-making, but instead provide feedback for **model updates**.

- **Random Generation**:
  The true probabilities are **randomly generated** within a defined range, such as **0.05 to 0.1**. This means that for each day, there is a random probability within this range that determines the chance of an email being opened. For example, a true probability of **0.07** for a particular day means that the chance of the email being opened on that day is 7%.

- **Why Simulated?**
  These probabilities are not based on real-world data but are used for simulation purposes to test how the model learns over time. By using random values, the model faces a level of uncertainty and unpredictability, similar to real-world scenarios where customer behavior is dynamic.

4.2 Reward Generation

In this project, **Reward Generation** refers to the process of determining whether an email was **opened** (success) or **ignored** (failure) during each trial. The outcome of each trial is represented by a **binary value**:

- **Success = 1** (email opened)
- **Failure = 0** (email not opened)

- **Purpose**:
  The **rewards** generated in this way are crucial for the model's learning process. They serve as **feedback** for the agent, telling it whether the action it took (sending an email on a particular day) was successful. This feedback is used by the model to update its decision-making process, guiding it toward better choices in future trials. The rewards are also used to calculate **regret**, which helps evaluate the model's learning by comparing the agent's performance to the optimal solution.

- **Reward Calculation**:
  The reward for each trial is determined by comparing a **randomly generated number** to the **true probability** for the selected day.
  - If the random number is **less than** the true probability, the email is considered **opened** (success, reward = 1).
  - If the random number is **greater than** the true probability, the email is considered **ignored** (failure, reward = 0).

- **Feedback for Model**:
  These rewards act as a **feedback signal** that helps the model understand how well it is performing. The model uses this feedback to adjust its strategy, learning over time which days lead to the best outcomes. By tracking success and failure, the model improves its ability to select the best days for email campaigns.

4.3 Customer Segments and Email Categories

In this project, the concept of **Customer Segments** and **Email Categories** is crucial for tailoring email campaigns to maximize engagement. These two components are used to simulate the diversity in customer behavior and ensure that the model optimizes timing for different types of audiences and messages.

- **Customer Segments**:

  Customer segments represent different groups of customers who may have varying behaviors, preferences, and interactions with the emails. In this project, customers are divided into **three distinct segments** based on **demographic characteristics**.
  These segments are:

  - **Young Adults (18-25)**: This group represents a younger audience, likely to have different email engagement patterns compared to older age groups.
  - **Working Professionals (26-45)**: These customers typically have a busy schedule, and the optimal time for them to check their emails might differ based on work hours and lifestyle.
  - **Seniors (45+)**: This segment may have different email engagement behavior due to varying habits and preferences.

  Dividing the customer base into these segments allows the model to consider **different behaviors and preferences** for each group. For example, younger people might be more likely to open emails at certain times of the day, while working professionals may have a preference for receiving emails during lunch breaks or after work hours.

- **Email Categories**:

  Emails are categorized based on the content or purpose they serve. Different categories of emails might require different timing strategies, as customer responsiveness can vary depending on the type of message. In this project, the primary email categories include:

  - **Promotions**: These emails typically contain offers, discounts, or sales promotions. Customers might be more responsive to these emails at specific times, such as during holiday seasons or weekends.
  - **Discounts**: These emails focus on offering discounts or special deals. Timing is important to ensure the email reaches the customer when they are most likely to make a purchase.
  - **Newsletters**: Regular updates or informational content. The timing might depend on the customer's general interest in receiving updates, and these emails may perform differently than promotional emails.

  By incorporating **email categories**, the model can account for the fact that the optimal time to send a **promotional email** might differ from the best time to send a **newsletter**. This allows for a more **personalized and targeted approach**, ensuring that each type of email reaches the customer at the best possible time.

- **Importance of Segments and Categories**:

  The combination of **customer segments** and **email categories** allows the model to simulate a variety of scenarios and optimize email sending times based on both **who the customer is** and **what type of content** is being sent. This segmentation enhances the accuracy of the model, ensuring that it learns the best sending times for **specific groups of people** and **specific email types**, improving overall engagement and open rates.

CHAPTER 5

METHODOLOGY

5.1 Beta Distribution

The **Beta distribution** is a **probability distribution** commonly used in statistics, particularly in the context of **reinforcement learning**. It is used to model **uncertainty** and is especially useful when dealing with **probabilities**, such as the probability of success for a given action.

In the context of the project, the **Beta distribution** is employed to represent the **belief** about the success probability of sending an email on each day of the week. This belief is continuously updated based on the feedback (rewards) the model receives from each action, which helps refine the model's decision-making over time.

Key Characteristics:

- **Shape**: The Beta distribution can take various shapes depending on its parameters, **α (alpha)** and **β (beta)**. These parameters control the **shape** of the distribution:
  - **α (alpha)**: Represents the number of **successful trials** (emails opened) + 1.
  - **β (beta)**: Represents the number of **failure trials** (emails ignored) + 1.

  The shape of the distribution adjusts as the values of α and β change, representing the model's evolving understanding of the success probability for a given day.
- **Range**: The Beta distribution has a range between **0 and 1**, making it ideal for modeling **probabilities**. This range directly corresponds to the likelihood of success (for example, the chance of an email being opened).

Mathematical Formulation:
The Beta distribution is defined by the probability density function (PDF):

$$\frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}$$

Where:

- x is the probability of success (email being opened).
- α (no. of successes +1) and β (no. of failures +1) are the parameters of the distribution.
- B(α,β) is the Beta function, which normalizes the distribution.

As the agent gathers more feedback (rewards), the values of **α** and **β** are updated. The distribution becomes more peaked around the **true probability** of success for the given action (i.e., sending an email on a specific day).

Application in the Project:

1. **Initial Beliefs**:
   Initially, the model doesn't know the best day to send emails. The Beta distribution starts with values of **α** and **β** set to 1, representing a uniform belief across all days. This implies that the model is equally uncertain about the likelihood of success on any given day.
2. **Updating the Distribution**:
   After each trial (whether an email is opened or ignored), the Beta distribution is updated:
   - If the email is opened (success), **α** is increased by 1.
   - If the email is ignored (failure), **β** is increased by 1.

   As these parameters are updated, the shape of the **Beta distribution** changes. Initially, when there is very little data (feedback), the distribution is **wide** and uncertain, meaning the model is unsure about which day is best for sending emails. But as the agent collects more rewards (successes and failures), the distribution **narrows** and becomes more **peaked** around the true probability of success for that specific action (i.e., sending an email on a particular day).

- **Peaked Distribution**:
  When the distribution becomes **more peaked**, it means the model is becoming more confident about the true probability of success for that day. If the true probability of success for a day is, say, **0.07** (7% chance of being opened), after several trials and rewards, the Beta distribution will "learn" this probability and concentrate around it. The agent will become more confident in choosing that day, especially if it has been successful more often than others.

- **Convergence**:
  As more data is gathered, the **α** and **β** values converge toward the **true probability** of success for that day. The agent is continuously refining its **belief** about the likelihood of success for each day, and the Beta distribution becomes **more focused** around this true probability, leading to better decision-making.

3. **Decision-Making**:
   For each trial, the model samples from the Beta distribution to determine which day to choose. Days with higher probability estimates (where **α** is significantly greater than **β**) are more likely to be selected, as they are considered more successful based on past experiences.

4. **Balance Between Exploration and Exploitation:**
   The Beta distribution helps manage the exploration vs. exploitation trade-off. When the distribution is wide (uncertain), the model is more likely to explore different days (exploration). As it becomes more confident in its estimates, the model will exploit the best-performing day (exploitation).

5.2 Thompson Sampling Algorithm

Thompson Sampling is a popular algorithm in **Reinforcement Learning (RL)** that helps make decisions in uncertain environments. It is widely used in **multi-armed bandit problems**, where the goal is to maximize the reward by choosing the best option from a set of possible actions. The key advantage of Thompson Sampling is its ability to **balance exploration** (trying out different options) and **exploitation** (focusing on the best-known option), ultimately leading to the best long-term outcome.

How Thompson Sampling Works:

Thompson Sampling is based on the **Bayesian approach**, where it maintains a **probabilistic belief** about the success or failure of each action. Following how it works in the project:

1. **Initial Setup**:
   - Each day (Sunday to Saturday) is considered an **action** or **option**. Initially, the model has no knowledge about the likelihood of an email being opened on any specific day. Therefore, the model starts with **uniform priors** (initial assumptions), often represented by Beta distributions with **α = 1** and **β = 1**, indicating maximum uncertainty.
2. **Exploration and Exploitation**:
   - In each trial, Thompson Sampling samples a probability from the Beta distribution for each day. This sample represents the **model's belief** about the success probability for each day.
   - The model then chooses the day with the **highest sampled probability** (exploitation), but sometimes it will choose a day with a **lower probability** (exploration) to gather more information about it.
   - This **exploration-exploitation trade-off** allows the model to **discover new strategies** while still optimizing for the best-performing days as more data is collected.
3. **Updating the Belief**:
   - After each trial, the model observes the outcome (whether the email was opened or ignored). This outcome is used to **update the Beta**

distribution for the selected day:

    - If the email is opened (success), the model **increases α** for that day by 1.
    - If the email is ignored (failure), the model **increases β** for that day by 1.

  - The updated Beta distribution represents the model's **refined belief** about the true probability of success for that day.

4. **Iterative Process**:
    - Over time, as more trials are conducted, the **Beta distributions** become more concentrated around the true probabilities of success for each day. The model **focuses more on the days with higher probabilities**, reducing exploration and maximizing exploitation.

Advantages of Thompson Sampling:

- **Efficient Decision Making**: Thompson Sampling balances the need to try new days (exploration) with the need to focus on the best day (exploitation), leading to faster convergence to optimal solutions.
- **Adaptive**: The model can continuously **adapt** to new data and changing customer behaviors by updating its beliefs with each trial.
- **Simple and Effective**: Despite its simplicity, Thompson Sampling is highly effective at finding the optimal solution over time.

5.3 Exploration vs Exploitation Trade-Off

One of the most critical aspects of **Reinforcement Learning (RL)** is the need to balance **exploration** and **exploitation** during decision-making. This trade-off determines how the model learns and adapts over time. **Thompson Sampling**, the algorithm used in the project, is designed to handle this trade-off effectively.

What is the Exploration vs. Exploitation Trade-Off?

1. **Exploration**:
    - Exploration involves **trying out different options** (days of the week in your project) to gather more information about their performance.
    - It helps the model learn about days that have been less tested and might have untapped potential.
    - Example: Testing a less frequently chosen day like Sunday to determine whether it might have a higher success rate than previously observed.
    -
2. **Exploitation**:
    - Exploitation focuses on **selecting the best-known option** based on current knowledge to maximize immediate rewards.
    - It ensures that the model takes advantage of the best-performing days to achieve optimal outcomes.
    - Example: Consistently choosing Thursday if it has been observed to have the highest email open rate so far.

The Importance of Balancing Exploration and Exploitation

Balancing exploration and exploitation is essential for the model to achieve the best long-term performance:

- **Too Much Exploration**:
    - ➢ The model spends excessive time testing suboptimal days, delaying the discovery of the best-performing days.

  - This leads to lower rewards during the exploration phase.
- **Too Much Exploitation**:
  - The model focuses too early on a single day, risking **local optima** (settling for a day that seems good initially but is not the best overall).
  - It might miss opportunities to discover better-performing days.

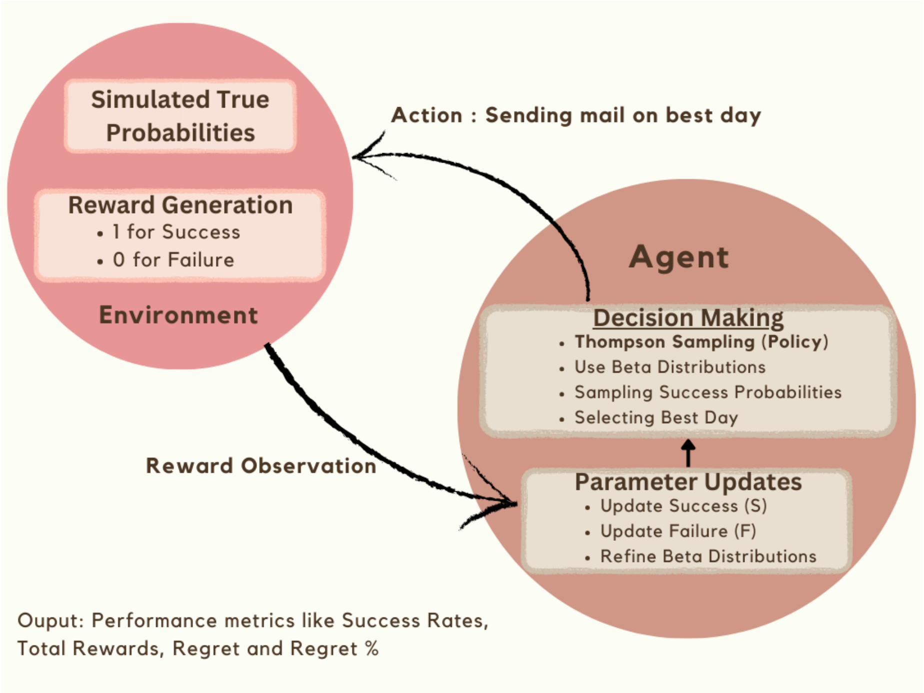
The ideal approach is a balance where the model explores new options to improve its knowledge while leveraging existing knowledge to maximize rewards.

How Thompson Sampling Manages the Trade-Off

Thompson Sampling handles this trade-off inherently through its probabilistic decision-making:

1. **Sampling from Beta Distributions**:
   - The algorithm samples a probability from the Beta distribution of each day, which reflects the model's belief about the likelihood of success for that day.
   - Days with higher uncertainty (wider Beta distributions) are more likely to be explored, while days with confident high success probabilities are more likely to be exploited.
2. **Dynamic Adjustment**:
   - Over time, as the model gathers more data and refines its beliefs, it naturally shifts from exploration to exploitation.
   - Early in the learning process, the model explores more, as all days have high uncertainty.
   - Later, as the Beta distributions converge around the true probabilities, the model increasingly exploits the best-performing days.

5.4 Model Learning and Performance Metrics


Simulated True Probabilities
Reward Generation
• 1 for Success
• 0 for Failure
Environment
Action : Sending mail on best day
Agent
Decision Making
• Thompson Sampling (Policy)
• Use Beta Distributions
• Sampling Success Probabilities
• Selecting Best Day
Parameter Updates
• Update Success (S)
• Update Failure (F)
• Refine Beta Distributions
Reward Observation
Ouput: Performance metrics like Success Rates, Total Rewards, Regret and Regret %

Model learning involves a **cyclical process** where the model interacts with a simulated environment, makes decisions based on current beliefs, receives feedback, and updates its parameters to improve over time. This process is essential for identifying the best days for email campaigns.

1. Data Preparation

- **Simulated True Probabilities**:
  - The likelihood of an email being opened is **randomly generated** within a defined range (e.g., 0.05 to 0.1). These probabilities are created for each **day of the week**, as well as for different **customer segments** and **email categories**.
  - They act as a **simulation of customer behavior**, providing the environment in which the model operates.
- **Reward Generation**:
  - After the model selects a day, a **random number** is generated.
  - If this random number is **less than the true probability** for the chosen day, the reward is **1** (email opened). Otherwise, the reward is **0**

(email ignored).

- These rewards simulate the outcomes of the email campaign and act as feedback for the model.

2. Decision Making

- The model uses **Thompson Sampling** to select a day for sending an email.
- Each day is represented by a **Beta distribution**, which encodes the model's belief about the success probability of that day.
- **Sampling from Beta Distributions**:
    - For each trial, the model samples a probability from the Beta distribution of each day.
    - It then selects the day with the **highest sampled probability** for that trial.
- This decision-making balances **exploration** (testing less certain days) and **exploitation** (choosing the best-performing day), allowing the model to adapt dynamically.

3. Updating Parameters

- After observing the outcome (reward), the model updates its **Success (S)** and **Failure (F)** counts for the selected day:
    - If the reward is **1** (success), the success count **S** for that day is incremented.
    - If the reward is **0** (failure), the failure count **F** for that day is incremented.
- These counts are used to update the parameters of the **Beta distribution** for each day:
    - The **α (alpha)** parameter represents number of successes + 1.
    - The **β (beta)** parameter represents number of failures + 1.
- Over time, the Beta distributions become more concentrated around the true probabilities, improving the model's belief and decision-making accuracy.

4. Performance Metrics

- To evaluate how well the model is learning, it tracks:
    - **Success Rates**: The ratio of successes (emails opened) to total trials

for each day.

- **Total Rewards**: The cumulative number of successes over all trials.
- **Regret**: The difference between the rewards achieved by the model and the rewards it could have achieved by always selecting the best day (with the highest true probability).
- **Regret Percentage**: Provides a normalized view of regret, showing how much performance has improved over time.

CHAPTER 6

MODEL ARCHITECTURE

6.1 Environment: Simulated Data and Feedback Loop

In the project, the environment plays a critical role by providing the simulated setting in which the agent (model) learns and makes decisions. It includes simulated data (true probabilities) and a feedback loop that helps the model refine its decisions over time.

Simulated Data: True Probabilities

- The environment includes **true probabilities** for email success, representing the likelihood of an email being opened on each day of the week (Sunday to Saturday).
- These probabilities are generated randomly within a defined range (e.g., 0.05 to 0.1) for different **customer segments** and **email categories**.
- **Purpose**:
  - ➢ These true probabilities simulate customer behavior, mimicking real-world scenarios where customer engagement depends on timing.
  - ➢ They act as a reference to determine whether an email is opened or ignored during each trial.

Reward Generation

- The **feedback loop** in the environment is based on the **reward system**, which evaluates the success or failure of each action (sending an email on a specific day).
  - ➢ **Reward = 1**: If the randomly generated probability (representing customer behavior) is **less than** the true probability for the chosen day, it indicates success (email opened).
  - ➢ **Reward = 0**: If the random probability **exceeds** the true probability, it

indicates failure (email not opened).

- **Purpose**:
  - ➢ These rewards provide feedback to the model, helping it refine its understanding of which days are more successful for sending emails.

Feedback Loop for Model Learning

- The feedback loop is essential for the **model's learning process**. It ensures that:
  1. The agent receives **immediate feedback** (reward) for its decision.
  2. This feedback is used to **update the Beta distributions** for the chosen day, refining the model's belief about success probabilities.
  3. Over time, the model learns to focus on the days with higher success probabilities, improving its decision-making.

Model Evaluation

- The rewards generated during trials are also crucial for **evaluating the model's performance**.
  - ➢ By tracking cumulative rewards, success rates, and regret, the project measures how effectively the model is learning and converging toward optimal decisions.

6.2 Decision-Making Process

In the project, the decision-making process revolves around the use of Thompson Sampling, a probabilistic algorithm that leverages Beta distributions to choose the optimal day for sending emails.

Using Beta Distributions for Decision-Making

- **Beta distributions** are used to model the agent's **belief** about the likelihood of success (email being opened) for each day of the week.
    - ➢ Each day (Sunday to Saturday) has its own Beta distribution.
    - ➢ The shape of the Beta distribution is determined by two parameters:
        - ▪ **α (alpha)**: Represents the number of successes (emails opened) + 1.
        - ▪ **β (beta)**: Represents the number of failures (emails ignored) + 1.
- **Sampling Probabilities**:
    - ➢ During each trial, the model samples a probability from the Beta distribution of each day.
    - ➢ These sampled probabilities represent the model's **estimated likelihood of success** for that day based on past observations.
    - ➢ The day with the **highest sampled probability** is chosen as the optimal day for that trial.

Balancing Exploration and Exploitation

- **Exploration**:
    - ➢ The model occasionally selects days with higher uncertainty (wider Beta distributions), even if they don't currently have the highest estimated probability of success.
    - ➢ This helps the model gather more information about underexplored days.
- **Exploitation**:
    - ➢ The model frequently chooses days with higher sampled probabilities (narrow, peaked Beta distributions), focusing on the days it has

learned are more successful.

  - This ensures the model maximizes immediate rewards by leveraging its current knowledge.

The balance between exploration and exploitation ensures that the model not only discovers the best-performing days but also capitalizes on them to achieve the highest long-term rewards.

Updating Beta Distributions

- After each trial, the model observes the **reward** (success or failure) and updates the Beta distribution for the selected day:
  - If the email was opened (**success**), the **success count (S)** for that day is incremented by 1.
  - If the email was ignored (**failure**), the **failure count (F)** for that day is incremented by 1.
- These updates refine the Beta distribution, making it more concentrated around the **true probability** for that day.
- Over time, the model becomes more confident in its decision-making, as the Beta distributions align more closely with the true probabilities.

6.3 Output

The outputs of the project include **total observed rewards**, **estimated probabilities**, and **cumulative regret**, which provide a comprehensive evaluation of the model's performance.

1. Total Observed Rewards

- **What It Is**:
    - The total observed rewards represent the cumulative number of successful outcomes (emails opened) across all trials.
    - It reflects how well the model performed in selecting the best days for sending emails.
- **In the Project**:
    - For each trial, if the email is opened, the reward is **1**; otherwise, it is **0**.
    - These rewards are summed up over all trials to calculate the **total rewards**, which directly indicate the effectiveness of the model's decision-making.

2. Estimated Probabilities

- **What It Is**:
    - The estimated probabilities represent the model's belief about the likelihood of success (email being opened) for each day.
    - These are derived from the Beta distributions, which are updated after each trial based on observed successes and failures.
- **In the Project**:
    - Over time, as the model learns from more trials, the estimated probabilities converge toward the **true probabilities** for each day.
    - The narrowing of Beta distributions around the true probabilities indicates the model's improved confidence in its estimates.

3. Cumulative Regret

- **What It Is**:
    - Cumulative regret measures the **difference between the rewards the**

model actually achieved and the **rewards it could have achieved** if it always chose the best-performing day (the day with the highest true probability).

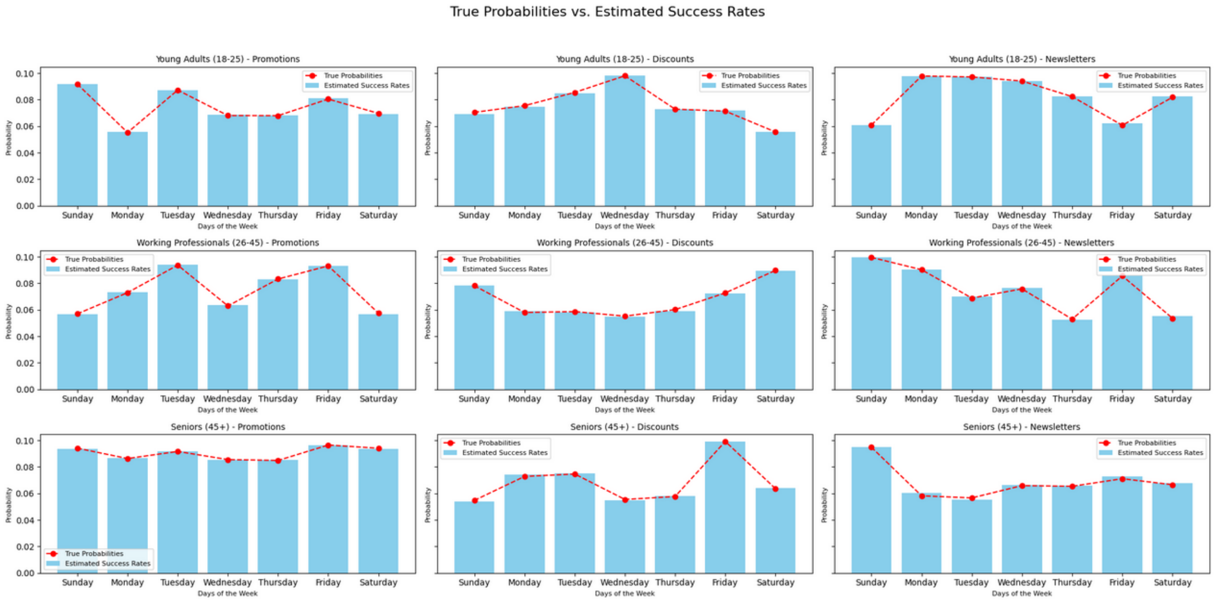
  - ➢ It highlights how much potential reward was lost due to exploration and imperfect decisions during the learning process.
- **In the Project**:
  - ➢ **Regret = Optimal Reward - Observed Reward**:
    - ▪ The optimal reward is calculated as the product of the highest true probability, the number of trials, and the number of customers.
    - ▪ Observed rewards are the cumulative rewards obtained by the model.
  - ➢ **Cumulative Regret** increases in absolute value over more trials because the model spends some time exploring less optimal days early in the learning process.

4. Regret Percentage

- **What It Is**:
  - ➢ Regret percentage provides a normalized view of cumulative regret by expressing it as a percentage of the optimal reward.
  - ➢ It offers a more intuitive measure of the model's learning efficiency, especially when comparing performance across different numbers of trials or settings.
- **In the Project**:
  - ➢ As the number of trials increases, the **regret percentage decreases** because the model learns and converges toward the optimal decision.
  - ➢ This demonstrates the model's ability to improve and minimize lost opportunities over time.

CHAPTER 7

RESULTS AND DISCUSSION



7.1 Estimated vs. True Probabilities

- **Result:**
  As the number of trials increases, the estimated probabilities for each day (Sunday to Saturday) become closer to the true probabilities. This demonstrates the model's ability to learn and refine its understanding of the optimal days for email campaigns over time.
- **Reason:**

1. **Learning Through Feedback**:
   - The model starts with an initial, uncertain belief about the success probability of each day, represented by uniform Beta distributions.
   - With each trial, the model observes whether the email was opened (success) or ignored (failure). This feedback is used to update the **success (S)** and **failure (F)** counts for the chosen day, refining the Beta distribution.
2. **Impact of More Data**:

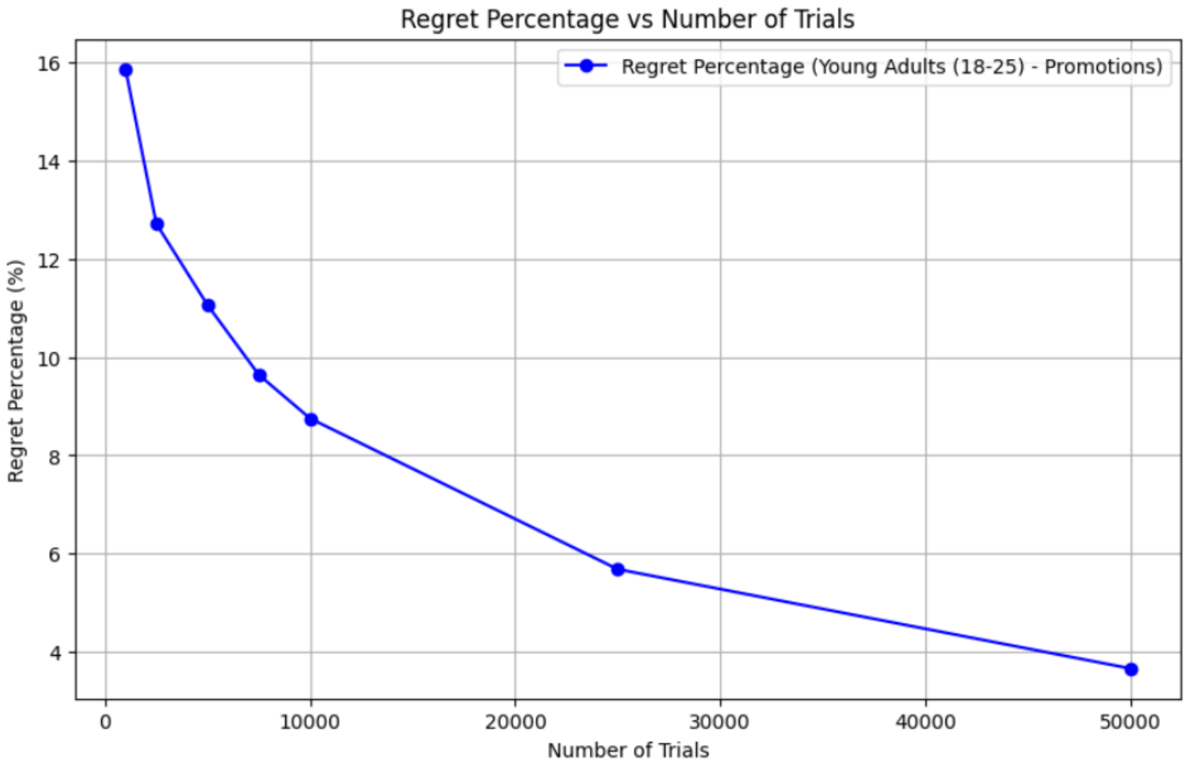
   - As more trials are conducted, the model gathers more feedback for each day. This increased data leads to more **precise updates** to the Beta distributions.
   - Over time, the Beta distributions for each day become **narrower and more peaked**, concentrating around the true success probabilities.

3. **Improving Accuracy Over Time**:
   - Early in the learning process, the model explores all days, leading to wider variability in its estimates.
   - With more trials, the model shifts toward exploiting the best-performing days, improving the accuracy of its estimated probabilities.
   - This iterative process ensures that the difference between the **true probabilities** (customer behavior) and the **estimated probabilities** (model's belief) decreases over time.

7.2 Regret Analysis: Absolute and Percentage Regret

Result 1: Absolute Value of Regret Increases with the Number of Trials

- **Observation:**
  - The **absolute value of regret** grows larger as the number of trials increases. This is because regret is a **cumulative measure** that sums up the difference between the **optimal rewards** (if the best day had always been chosen) and the **actual rewards** achieved by the model.
- **Reason:**
  - Early in the learning process, the model makes **sub-optimal decisions** as it explores different days to gather information. These exploratory decisions result in missed rewards compared to what could have been achieved by consistently selecting the best day.
  - As the number of trials increases, these early sub-optimal choices continue to contribute to the cumulative regret, causing it to grow over time.



Result 2: Regret Percentage Decreases with the Number of Trials

- **Observation:**
  - ➢ Despite the increase in absolute regret, the regret percentage (regret normalized by the optimal reward) decreases as the number of trials grows.
- **Reason:**
  - ➢ Over time, the model learns to identify and exploit the optimal day more frequently by refining its Beta distributions.
  - ➢ With more trials, the number of optimal decisions increases, reducing the relative impact of early sub-optimal decisions.
  - ➢ As a result, the proportion of regret compared to the total potential reward decreases, even though the absolute value of regret continues to grow.

7.3 Model Performance Across Trials

The performance of the model improves progressively as it undergoes multiple trials.

Performance Indicators

1. **Improved Success Rates**:
    - As the number of trials increases, the model's success rate (the proportion of opened emails) improves.
    - This indicates that the model is increasingly identifying and selecting the best days for email campaigns, maximizing rewards over time.
2. **Convergence of Estimated Probabilities**:
    - The estimated probabilities for each day gradually converge toward their respective true probabilities.
    - This convergence reflects the model’s ability to refine its beliefs and make decisions with greater accuracy as more data is observed.
3. **Reduction in Regret Percentage**:
    - The model demonstrates a decline in regret percentage as it learns to choose the optimal day more frequently.
    - This decrease shows that the model is minimizing lost opportunities relative to the total potential reward, a critical measure of its learning efficiency.

Learning Process Across Trials

- **Early Trials**:
    - During the initial phase, the model focuses on **exploration**, testing different days to gather sufficient data about their performance.
    - Success rates and estimated probabilities are lower, and regret is higher due to the lack of prior knowledge.
- **Later Trials**:
    - As trials progress, the model shifts toward **exploitation**, focusing more on days with high success probabilities.
    - This results in higher success rates, improved accuracy in estimated probabilities, and reduced regret percentage.

CHAPTER 8

CHALLENGES

8.1 Balancing Exploration and Exploitation

One of the key challenges stems from the inherent trade-off between **exploration** and **exploitation** in **Reinforcement Learning (RL)**. This balance is essential for the model to learn effectively and avoid suboptimal outcomes.

- **Exploration**:
  - ➢ The model needs to **explore different days** to gather information about their success probabilities.
  - ➢ Insufficient exploration may prevent the model from discovering the best-performing day, especially if some days initially appear less favorable.
  - ➢ **Risk**: If the agent focuses excessively on exploration, it spends too much time testing suboptimal days, resulting in reduced overall rewards.

- **Exploitation**:
  - ➢ The model must also **exploit the best-known day** to maximize immediate rewards.
  - ➢ Overemphasis on exploitation leads to the risk of settling for a day that seems good based on limited information but is actually suboptimal (**local minima**).
  - ➢ **Risk**: The model might miss out on identifying better-performing days due to its reluctance to try less-tested options.

How Thompson Sampling Addresses This Challenge

1. **Dynamic Balance**:
    - Thompson Sampling inherently balances exploration and exploitation by sampling from **Beta distributions**.
    - Days with higher uncertainty (wider distributions) are more likely to be explored, while days with higher success probabilities (narrow, peaked distributions) are exploited.

2. **Gradual Learning**:
    - Over time, as the model refines its Beta distributions through observed rewards, it naturally reduces exploration and focuses more on exploitation.
    - This approach ensures that the agent avoids local minima and converges toward the optimal solution.

8.2 Computationtal Complexity

As the scope of the project expands, particularly with increasing **trials** and **customer segments**, the computational demands also increase.

- **Number of Trials**:
  - ➢ The model learns by performing multiple trials, where each trial involves selecting a day to send an email and updating the Beta distributions based on the observed outcomes (whether the email was opened or ignored).
  - ➢ As the number of trials increases, the model needs to perform more **sampling** and **updating** steps for each day (Sunday to Saturday), which naturally increases the computation time.

- **Customer Segments**:
  - ➢ Your project involves multiple **customer segments**, such as **young adults**, **working professionals**, and **seniors**.
  - ➢ Each segment requires its own set of **probability distributions** and updates, adding to the complexity. For example, for each customer segment, the model needs to perform the same calculations (sampling from Beta distributions, updating successes/failures) for each email category and trial.

- **Model Scalability**:
  - ➢ As the number of trials and segments grows, the number of calculations required to maintain and update the Beta distributions increases exponentially. This can lead to longer computation times and increased memory usage, making the model harder to scale efficiently

Impact:

- The increased number of **trials** and **customer segments** can lead to:
    - **Longer training times**, as the model has to iterate over more data and update distributions more frequently.
    - **Higher resource consumption**, requiring more memory and computational power to handle the increasing complexity.
    - Potential **delays in performance evaluation**, as tracking the model's progress across many trials and segments takes more time.

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 Conclusion

We summarize the key findings and highlight the effectiveness of Thompson Sampling in optimizing email campaign timings, as well as the insights gained through regret analysis.

1. Thompson Sampling Effectively Identifies Optimal Days for Email Campaigns

- **Thompson Sampling** was successfully applied in your project to identify the best days for sending emails to maximize customer engagement.
  - ➢ The model used **Beta distributions** to represent the agent's belief about the likelihood of success (email being opened) for each day.
  - ➢ Over time, as the model was exposed to more data and feedback, it **refined its beliefs** and focused on the days with the highest success probabilities, effectively identifying the optimal days for email campaigns.

2. Regret Analysis Shows Learning and Convergence Over Time

- **Regret analysis** provided a crucial insight into the model's learning process.
  - ➢ In the early stages, the model faced **higher regret** as it explored less optimal days to gather information.
  - ➢ However, as the model learned from more trials, **regret decreased** and the model gradually converged towards selecting the best-performing days.
  - ➢ This demonstrates that the model was able to **learn from its past actions** and improve its performance over time, ultimately reducing the loss in potential rewards.

3. Probabilistic Approaches Improve Decision-Making in Marketing

- By using **probabilistic methods** like Thompson Sampling, your project demonstrates how uncertainty in decision-making can be effectively modeled and used to make informed choices.
    - ➢ The ability to sample from **Beta distributions** allowed the model to balance **exploration** (testing new days) and **exploitation** (focusing on the best-performing days).
    - ➢ This probabilistic approach enhanced the model’s ability to optimize email send times, offering a more data-driven, adaptive strategy than traditional rule-based methods.

9.2 Scope for Future Work

This section outlines potential avenues for expanding and improving the project in the future. These enhancements aim to build upon the current model's capabilities and explore additional strategies for optimizing email campaign effectiveness.

1. Incorporating More Sophisticated Reinforcement Learning Techniques

- While Thompson Sampling has proven effective in balancing exploration and exploitation, future work could explore more **advanced reinforcement learning (RL) algorithms.**
- Advanced RL techniques can handle more complex scenarios, such as dynamically adjusting strategies based on evolving customer behavior or optimizing multiple objectives (e.g., maximizing open rates while minimizing campaign costs).
- By integrating more advanced RL methods, the model can achieve even greater precision and adaptability in decision-making**.**

2. Expanding Metrics for Interaction Effectiveness

- Currently, the model evaluates performance using metrics like **success rates**, **rewards**, and regret. Future work could expand these metrics to include:
  - ➢ **Click-Through Rates (CTR):** Measures the percentage of email recipients who click on links within the email, offering deeper insights into engagement.
  - ➢ **Conversion Rates**: Tracks the percentage of recipients who complete desired actions (e.g., purchases or sign-ups) after interacting with the email.
- These additional metrics would provide a more **comprehensive evaluation** of the campaign's effectiveness, beyond just email open rates, allowing for better optimization of marketing strategies**.**

3. Optimizing Email Open Times for Broader Customer Segments

- The current project focuses on predefined customer segments (e.g., young adults, working professionals, seniors). Future enhancements could incorporate:
    - **Demographic Data:** Tailoring email timing strategies based on factors like age, location, or gender.
    - **Behavioral Data:** Leveraging insights such as previous engagement patterns, shopping habits, or time spent on the platform to refine targeting.
- By expanding to include more **granular segmentation** and personalization, the model can better address diverse customer needs, further enhancing engagement and campaign success.