Optimizing Web Forms using A/B Testing and Reinforcement Learning

CSC2558 - Project Report

Parul Saini, Pranav Gupta, Weiming Ren
(parul.saini@mail.utoronto.ca) (pranav@cs.toronto.edu) (weiming.ren@mail.utoronto.ca)

Abstract

In this study, we have performed a series of experiments to test various approaches to optimize the Web Forms by improving user experience. The web forms are developed by incorporating Design Thinking principles. We then conduct a live Randomized A/B experiment and present a group of people with two versions of the form randomly. Furthermore, we simulate Epsilon Greedy and Thompson Sampling on the data gathered from our experiment to evaluate the likelihood of a user choosing to fill an Optimized Form over its Unoptimized alternative. We comprehensively compare the performance of these algorithms on our data and conclude that Beta Bernoulli Thompsons Sampling performs the best, with least regret therefore establishing the superiority of probabilistic approach.

1. Introduction

1.1 Background

Web forms are vital for information gathering and communication between a customer and website owner. Customer feedback, registration and subscription are some of the use cases for web forms, hence they need to be interactive and engaging. However, most website owners and companies struggle with getting quality feedback and user inputs through web forms as they tend to be lengthy and tedious. Our goal was to apply a design thinking approach around building a web form that is appealing to the user and the other which is a generic form and test our hypothesis using Randomized A/B testing and Multi-arm Bandit algorithms. We intend to improve the user experience by optimizing the web form using these experiments such that a developer/company can leverage this technique to get maximum response.

1.2 Related Work

Forms are used in websites for improving business and user experience, as well as for various purposes like surveys, collecting feedback, etc. Length, design and many other factors prove to be a bottleneck in the process of collection of user inputs by the means of forms.

The aim is to select the best form to put into a website so that more users will fill it. In [2] the first approach is a Randomized A/B test that is performed for three forms with distinct designs and

distributed uniformly among users. In the second approach, the Adaptive Epsilon Greedy algorithm was used to test forms of three versions with a set of users. It is concluded that the Adaptive epsilon greedy algorithm finds the best form quicker and more efficiently than A/B test.

Adaptive Epsilon Greedy algorithm [4] – a form of multi-armed bandit algorithm that is used in [2] selects the best possible option out of the multiple options keeping profitability and efficiency into consideration. The Adaptive Epsilon Greedy algorithm in Reinforcement Learning is based on the classic ε-greedy method, where the state of the agent is modified based on some stimulus received with a goal to maximize the reward. This dynamic system helps attain a balance between exploration and exploitation [3]. Using the approach suggested in [5], we design our simulations for real-world data.

Our Contributions: We design two forms, A and B, where A is a standard yet suboptimal form whereas B incorporates design thinking principles [1] to improve user experience. We conduct a Randomized A/B Experiment to gather responses from real-time users for our two forms and use the data to simulate reinforcement learning algorithms. We replicate the two techniques discussed in [2] for the two forms. In addition to this, we implement a simulation with Beta-Bernoulli bandit and Thompson Sampling in order to compare the results obtained with the first two techniques and identify the best approach [6] . The comparative analysis of the three techniques in reference to optimizing web forms has not been done before which makes our study novel.

2. Design Thinking for Web Form Creation

We designed two web forms A and B that required users to share about ‘Life in a Pandemic’. The design of Form A is standard but suboptimal such that it contains long and winding questions, vague choices and some other common pitfalls of web form design. We then designed Form B, on which we applied design thinking knowledge to optimize the user experience of filling this form. To achieve this, we conducted user interviews to explore the common issues people face while filling the forms, and then did prototyping to improve Form B iteratively.

Based on user interviews we received the following feedback:

1. People have varied interests such as movies, food, stock market, etc. Our goal was to curate forms on a topic that everyone could relate well with, hence ‘Life in a Pandemic’ seemed to be a good choice.
2. Given a choice, checkboxes and MCQ were the most preferred question type that the users would like to see in a form. On further research some users were interested to answer with short answers if compulsory/imperative to the research.
3. While asking the users the source of forms, most users said they shall always prefer a feedback form from a trusted source than from an unknown/untrusted source.
4. The average time preferred for filling a feedback form was observed to be 5-6 minutes based on the interviews conducted.

5. Finding the most suitable time to send the form is as crucial to achieving maximum user engagement, as the contents of the form. Based on the user interviews we found that evening is the most suitable time for filling a survey/feedback web form.
6. In order to understand the user better we gave them a choice on which type of form they shall not prefer out of Time consuming / Lengthy / Boring / Irrelevant, to which most users avoid lengthy and time consuming forms. It was interesting to note that users would fill a short boring/irrelevant form as opposed to lengthy interesting forms.
7. Users prefer filling non-mandatory fields if they are MCQ.
8. While understanding the mandatory fields that the users dislike the most, we came across various answers such as ‘make an account’, ‘MCQ not having N/A option’, ‘unnecessary/irrelevant to form’. The most disliked question was the one with a long paragraph as a field to answer.

The user interviews gave us a sense of what the users wanted and how we could improve the design of our form as delineated above. While conducting these interviews there were some issues that we faced such as:

1. Medium: There were 3 mediums through which the user interviews were conducted, that were google forms, phone call and video call. Although the forms were the fastest method to conduct the interviews, it did not give the opportunity to converse and capture user thoughts like in voice/video call.
2. Time: Interviews over video call were observed to last longer as opposed to the form and voice over call medium. This was mainly because users were more interactive and had more time to express themselves.
3. Documentation: Google form was the most efficient way to document as it automatically saved the answers in a spreadsheet which is downloadable. Whereas the voice and video call took more effort. We made notes during the interview and documented after carefully listening to the recordings (recorded after taking user consent).

In Form B we addressed the most common issues faced by people. Both the forms consisted of 20 questions in which the last question required people to rate the form from 1-5 based on ease of filling. We also added a few demographic questions to understand our audience better such as their age, gender, geographic location etc.

3. Randomized A/B Testing Web Form and Data Preparation

We performed Randomized A/B Testing as shown in Fig. 1 to assign a group of users with two form variants. We then collect the responses obtained using the experiment to simulate the algorithms in the subsequent sections. In our study, we refer to the optimized form as Form B/Form 1 and unoptimized form as Form A/Form 0. The idea is that Form B is a better variant, and we were able to establish this thought through our A/B Testing Experiment and in the following sections with Multi-armed Bandit and Thompsons Sampling.

Fig. 1. Design of A/B Experiment

3.1 Randomized A/B Testing

As shown in Fig. 2, the experiment presented users randomly with Form A which is a standard, straightforward web form without any bells and whistles and Form B which considers the User Experience and Design Thinking approach as discussed in section 2.

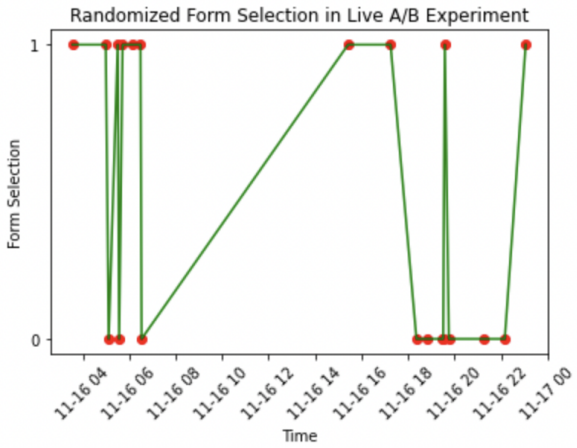


Fig. 2. Visualization of randomized allocation of the two forms as a function of time

The experiment was live for two weeks in order to get user inputs as highlighted in the next section. We used a paid platform - TypeForm to design our forms and collect the responses due to its interactive UI and better flexibility at capturing form statistics. Fig. 3 shows the insights corresponding to user interaction with each of the forms.

	#Views	#Starts	#Submissions	Completion Rate	Avg Time to Complete
Form A	128	104	45	43.3%	13.03
Form B	108	82	55	67.1%	05:04

Fig. 3. Form Insights

We were able to reach out to more than 200 people out of which 100 users successfully submitted the forms. In line with our expectations, Form B had a higher completion rate and

lower average time to complete due to its ease of usability as it was designed keeping user feedback and design thinking principles in mind.

3.2 Dataset Preparation

For dataset preparation, we collected real-time input from users by running a Randomized A/B experiment. Based upon parameters like completion rate, submission rate of forms and other demographic information recorded in forms, we analyze the response of a certain group to the design of the form.

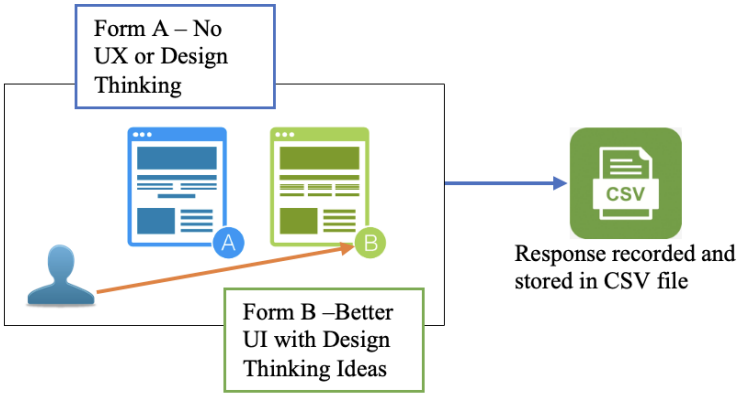


Fig 4. Recording User Responses Real-Time

3.3 Dataset Exploration and Visualization

The data we gathered from the Randomized A/B experiment was then analyzed for both the optimized and the unoptimized forms. In this section we shall explore the demographics of the users/participants on two parameters namely gender and occupation, we also compare the completion rate for both forms since the optimized form was simpler to answer and had more non-mandatory fields than the unoptimized form.

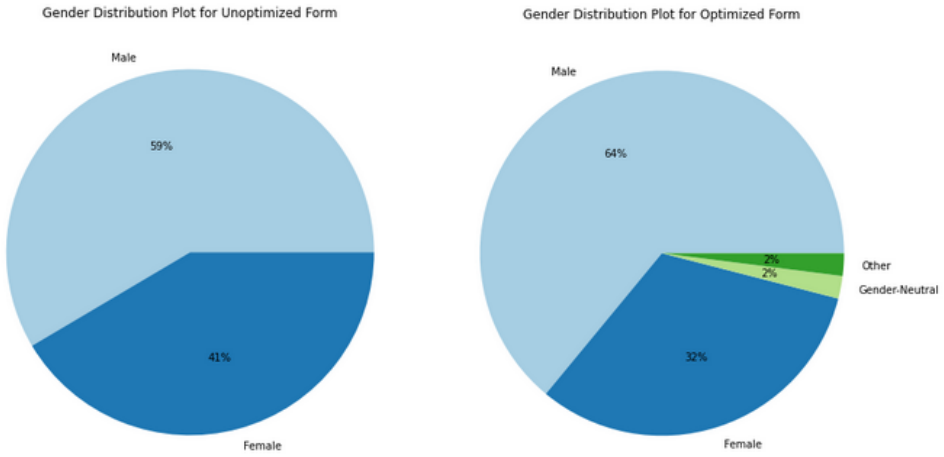


Fig. 5: Gender Distribution Plots for Optimized and Unoptimized Forms

In Fig. 5, we see gender distribution plots and it is observed that in the unoptimized form the users are either male or female, out of the 3 options given i.e. male, female and others. However with optimized form the choice given to users made it more flexible as the choices of pronouns given were more diverse such as sher/her, he/him, they/them, prefer not to say and other. With optimized form being more descriptive, it allows users to enter their gender/pronoun more appropriately.

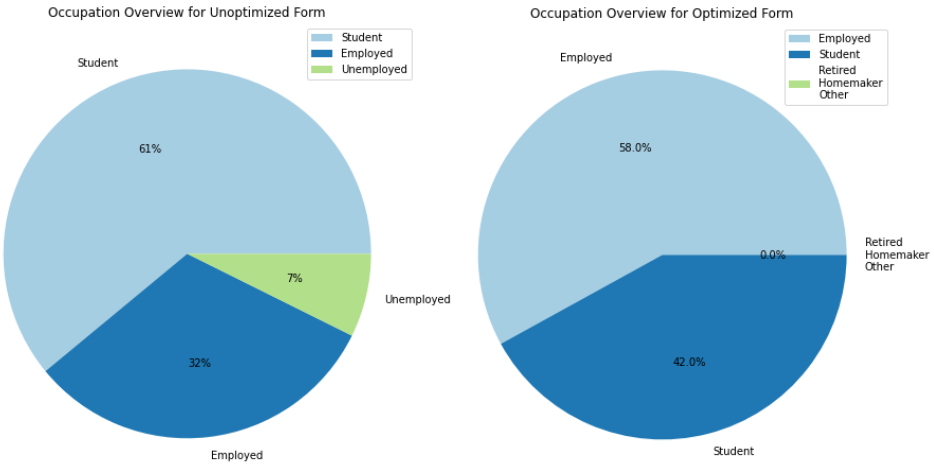


Fig. 6: Occupation Overview for Optimized and Unoptimized Forms

Further we explore the occupation of our participants, the unoptimized form consisted of a short answer field where the users typed their occupation which have been broadly classified as student, employed and unemployed based on the answers received. It can be observed from Fig. 6 that an MCQ based choice was provided to users in the optimized form, while most users who filled the forms were either students or employed, we believe that these fields can be explored in future research on a larger user testing.

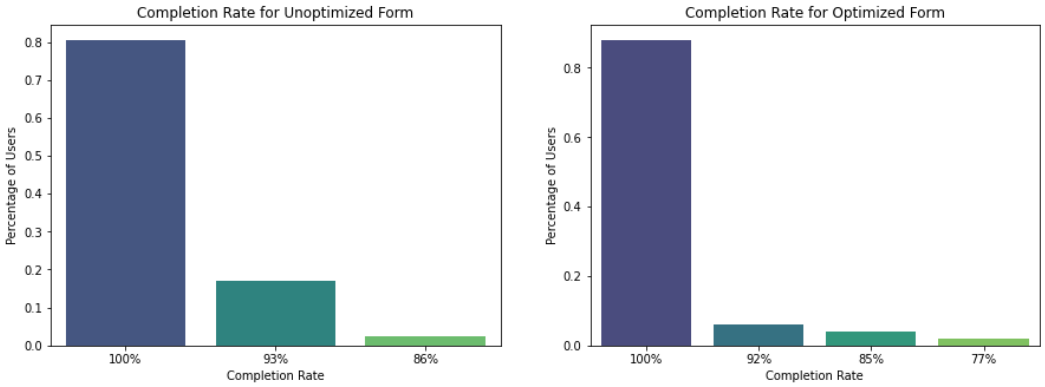


Fig. 7: Completion Rate for Optimized and Unoptimized Forms

Finally, we compare the completion rate for both forms i.e how many fields the participants filled, the plots show that the percentage of users completing the optimized form is 10% more than the unoptimized form. Interestingly, we find that participants completing about 93% of the form were more in the unoptimized version. This aligns with our assumption that most users fill the form

which was more optimized and easier to use, while filling most of the form but not completing the unoptimized version either due to time, length or difficulty to fill.

4.Multi-armed Bandit Approach for Optimization using Epsilon-Greedy Strategy

A multi-armed bandit solution is a ‘smarter’ or more complex version of A/B Testing that uses machine learning algorithms to dynamically allocate traffic to variations that are performing well, while allocating less traffic to variations that are underperforming. From the dataset generated in the previous section, we simulate a Multi-armed Bandit algorithm using Epsilon-Greedy Strategy where the state of the agent is modified based on some stimulus received with a goal to maximize the reward.

In Algorithm 1 [7], we assume a no-information prior for initializing our rewards and choose different values of ε to simulate the algorithm , here ε (epsilon) refers to the probability of choosing to explore, exploits most of the time with a small chance of exploring. The Choose_Form() function generates a random number between 0 and 1. If it’s greater than epsilon, it directs us to exploit the function. Otherwise, it directs us to the explore function. The Exploit() function instructs us to choose the form that has the highest cumulative reward (our “greedy” action) while the Explore() function instructs us to choose randomly among the two forms.

```
Algorithm 1 : ε - Greedy Multi Arm Bandit Algorithm:

Input: Different Values of ε , Cumulative Reward for Form A and Form B
Output: Form A or Form B
def explore():

    return random.choice(forms)

def exploit(cum_reward_formA ,cum_reward_formB):

    if cum_reward_formA > cum_reward_formB:
        return formA
    elif cum_reward_formB > cum_reward_formA:
        return formB
    else:
        explore()

def choose_form(ε ,cum_reward_formA ,cum_reward_formB):

    if random.random()> ε:
        return exploit(cum_reward_formA ,cum_reward_formB)
    else:
        return explore()
```

5.Optimization using Beta-Bernoulli bandit and Thompson Sampling

Beta-Bernoulli bandit is a type of multi-armed bandit that receives binary rewards for each of its arms. 'Beta' refers to the Beta distribution, which is used to parametrize the probability of an arm getting a reward. It is known that a Beta distribution of Beta(1 + α, 1 + β) models a Bernoulli random variable with α observed success and β observed failures.

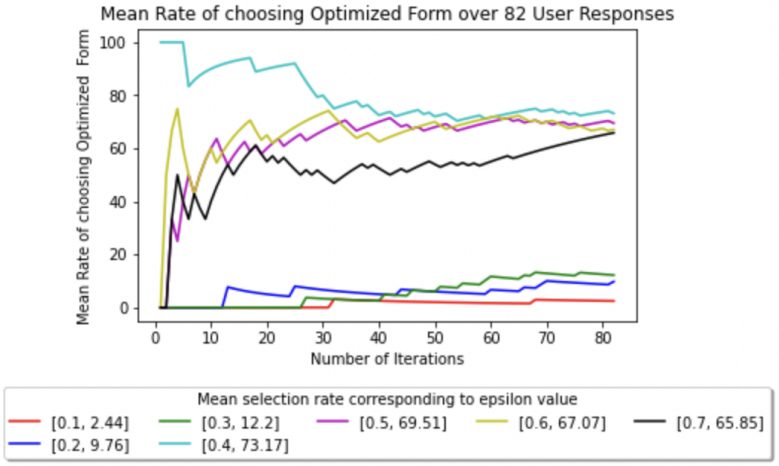
Similar to the epsilon-greedy algorithm as discussed in section 4, Thompson sampling is also a technique for dynamically adjusting the exploration and exploitation rate for the reinforcement learning algorithm and choosing the optimal arm from the bandit. The idea behind Thompson sampling is called 'probability matching' [8], which means that the probability of an arm being chosen by the algorithm is equal to its probability of being an optimal arm. As aligned with Algorithm 1, we assume a no-information prior for both optimized and unoptimized forms. For Beta-Bernoulli bandits, this refers to a Beta(1, 1) (uniform distribution) initialization for both arms. During each iteration, we first choose a form using choose_form() based on the posterior distribution of the forms after the previous round of iteration. After observing the reward for our choice, we then update the distributions using update_beta_distribution().

```
Algorithm 2 : Beta-Bernoulli bandit and Thompson Sampling:

Input: Posterior arm distributions for Form A and Form B. Both input
variables contain a list of [success, failure] observations.
Output: Form A or Form B


def choose_form(beta_formA, beta_formB):

    # draw from arm distributions
    val_formA = np.random.beta(1 + beta_formA[0], 1 + beta_formA[1])
    val_formB = np.random.beta(1 + beta_formB[0], 1 + beta_formB[1])

    return formA if val_formA >= val_formB else formB

def update_beta_distribution(choice, beta_formA, beta_formB):
    if (choice == formA):
        reward = sample_from_dataset(formA) # reward is either 0 or 1
        beta_formA[1 - reward] += 1
    else:
        reward = sample_from_dataset(formB) # reward is either 0 or 1
        beta_formB[1 - reward] += 1
    return beta_formA, beta_formB
```

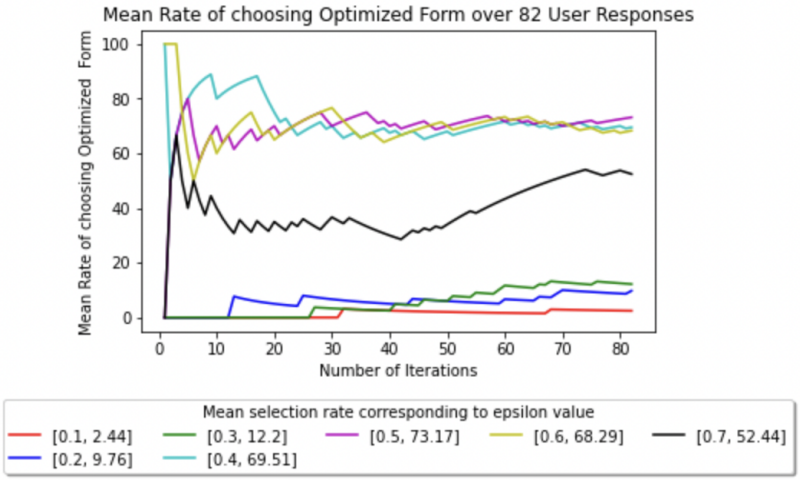
6. Results

6.1 Epsilon - Greedy Multi Arm Bandit Analysis

We consider this simulation with two rewards: (1) User rating as the reward and (2) Reward based on whether the user submitted the form or not. We try to understand the impact of considering different reward types on the results observed as shown in Fig. 8. Additionally, simulation (2) is performed for fair comparison with Beta Bernoulli - Thompson Sampling considered in the section 5. The observation was that both simulations achieve the mean rate of 73.17% but for different epsilon values. While simulation with user rating as reward archives maximum mean rate value for ε = 0.4 , simulation with user submission as reward achieves the same rate at ε = 0.5. Therefore tuning the hyperparameter ε is important for choosing optimal performing configuration.



(1) Simulation with user rating as the reward



(2) Simulation with user submission as the reward

Fig. 8. Mean Rate of Choosing Optimized form for different epsilon using different reward types

We further analyze the cumulative reward curve and regret plot for simulation (2) as shown in Fig. 9. Corresponding to ε=0.5 we observe the maximum cumulative reward and minimum regret as we saw maximum selection rate corresponding to the same epsilon value in Fig. 8. Based on our analysis we establish the fact that the epsilon-greedy algorithm continues to explore with probability ε and exploit the best known action with probability 1-ε to find optimal action and policy. Therefore too much exploration (high ε) and exploitation (small ε) doesn't yield optimal results in comparison to an intermediate value of ε.

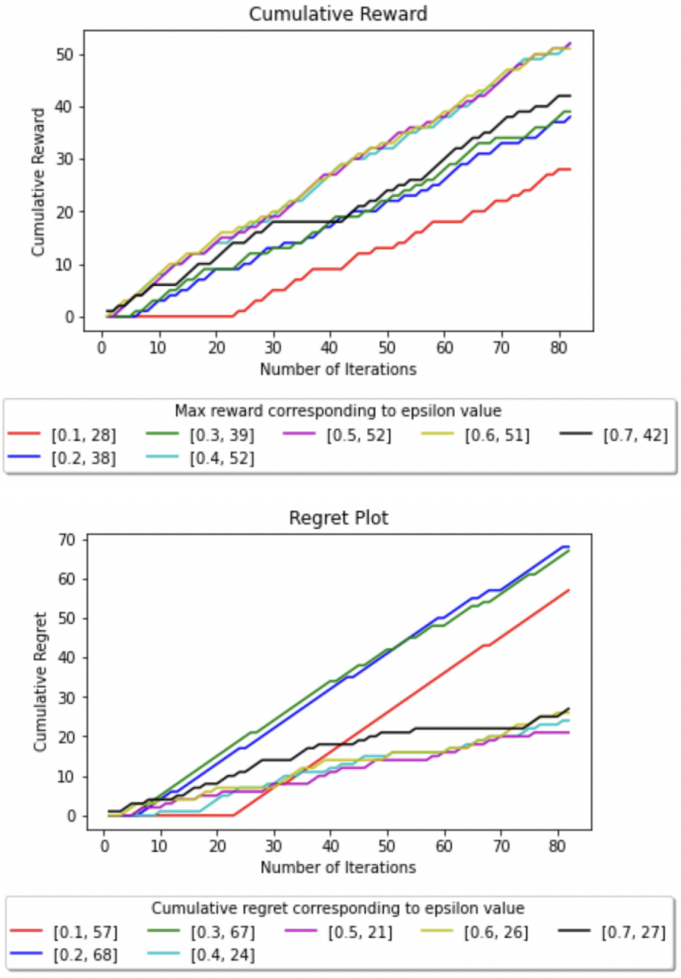


Fig. 9. Cumulative reward and Regret plot for different epsilon using user submission as the reward

6.2 Beta-Bernoulli bandit and Thompson Sampling Analysis

For the Thompson sampling simulation, we initialize the Beta distribution for both unoptimized and optimized form to Beta(1, 1) and run the algorithm for a total of 82 iterations. To understand the learning behavior of the Thompson sampling algorithm, for every 40 iterations, we plot the beta distribution for both unoptimized and optimized form. The distribution plots are shown in Fig. 10. below.

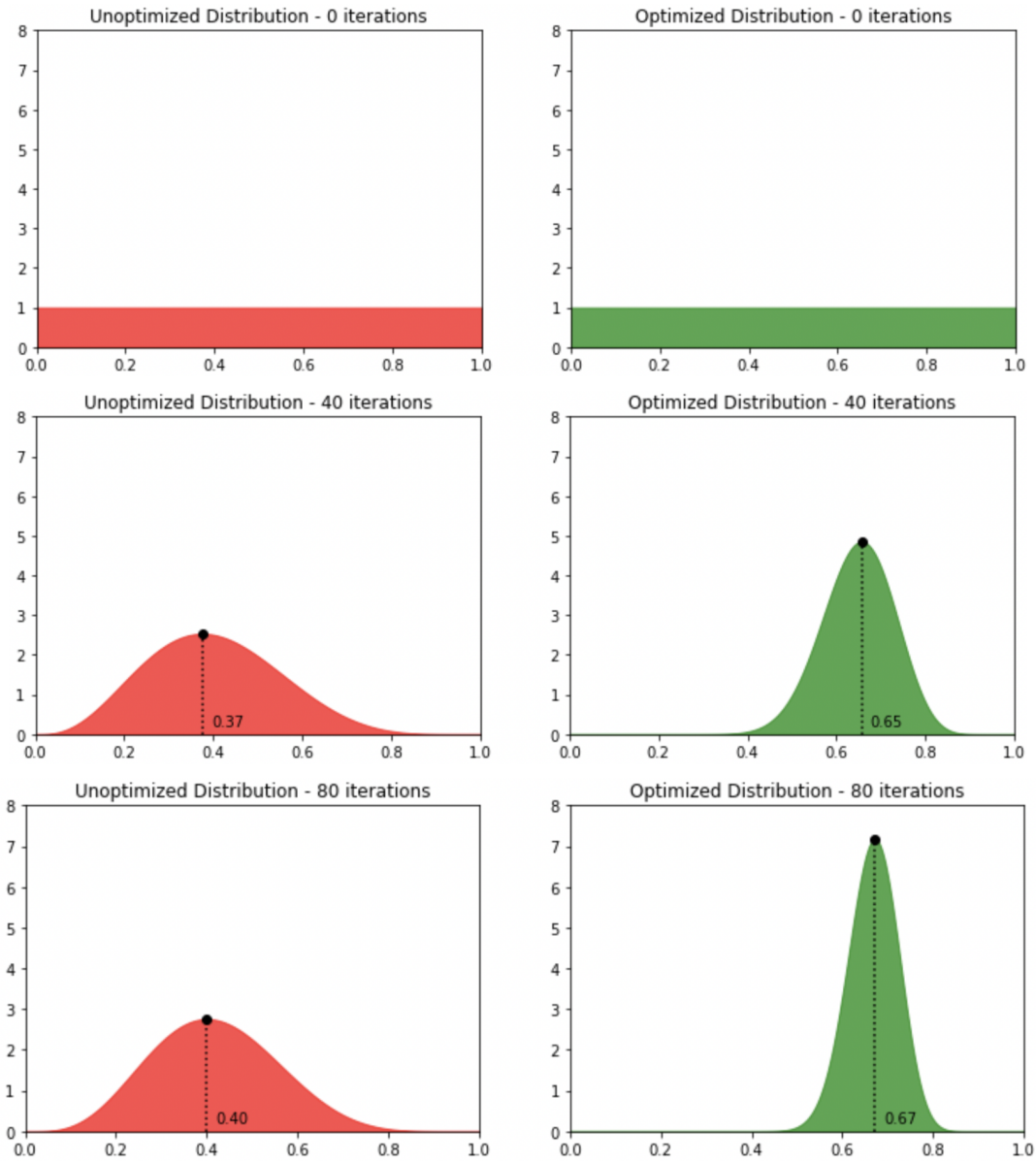


Fig. 10. Beta distribution for unoptimized and optimized forms during Thompson sampling iterations

As shown in Fig. 10, both forms start from the uniform distribution at the beginning of our experiment. Throughout the experiment, this distribution is updated based on the observation of the rewards coming from the simulation dataset. We can find that after 40 iterations, the difference between the optimized and unoptimized distributions is already significant. The optimized distribution has a peak value at 0.65, while the unoptimized distribution only has its peak value at 0.37. This indicates that when sampling from these two distributions, it is much more likely for the optimized value to be greater than the unoptimized value, thus the algorithm will choose the optimized form more often. When the experiment proceeds to 80 iterations, we find that although the peak location for both distributions does not change too much compared to the 40 iterations plot, the optimized distribution has a much higher peak value (>7), and the distribution becomes more leptokurtic. This indicates that as the observation for the optimized

form accumulates, the algorithm becomes more and more certain that the probability of getting a reward from the optimized form is around 0.67, which is aligned with the true probability of 0.671 as shown in Fig. 3. It should be noted that although the unoptimized distribution plot also shows a very similar peak value compared to true probability (0.4 to 0.433), we find that this value fluctuates significantly for different rounds of experiment when we conduct the same experiments repeatedly. This is because the unoptimized distribution is only updated for a limited time during the experiment as the algorithm quickly begins to focus on exploitation on the optimized form after a few iterations.

6.3 Comparative Performance Analysis

We compared the Randomized A/B Testing, Epsilon Greedy MAB and Beta Bernoulli Thompsons Sampling as shown in Fig 11. For Epsilon Greedy we choose the results corresponding to best performing epsilon value when reward is user submission for fair comparison with Beta Bernoulli Thompson's Sampling.

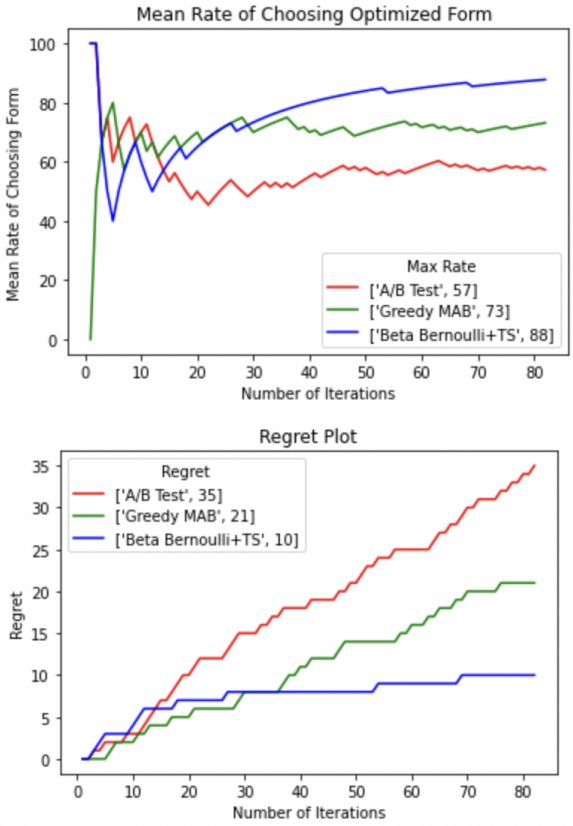


Fig 11 . a) Performance of different techniques for selecting optimized versions of the from b) Regret plot corresponding to each technique.

We observe that corresponding to A/B Testing the regret is highest while the cumulative reward is lowest. On the contrary, Thompson's Sampling exhibits minimum regret and maximum reward

suggesting which is even higher than Epsilon Greedy MAB. This is intuitive as ϵ-Greedy method will just stick with arm that yield good results. In the beginning, Thompson Sampling invested some time into getting good estimates before rejecting bad bandits and during that time ϵ-Greedy's usage of good but suboptimal arms will yield more rewards. However as number of iterations increase, Thompson's outperforms ϵ-Greedy as (1) ϵ-Greedy with low value of epsilon is much more susceptible to getting stuck with good but not optimal arm (2) ϵ-Greedy with high value of epsilon necessarily samples suboptimal arm more often This suggests a probabilistic approach is therefore superior [9]. If Thompson's Sampling is employed user's will be directed to the best choice available. Since Randomized A/B Testing doesn't take reward into consideration, it was expected to make random allocations of the two available alternatives and therefore the observed results are in line with our expectations.

7. Discussion and Conclusion

In this paper, we have presented two web forms that are designed by incorporating design thinking principles. We then collected user responses using randomized A/B testing and proposed a comparison study of two reinforcement learning algorithms epsilon-greedy and Thompson sampling on this real-world web form optimization problem. Our experimental results showed that both algorithms performed better than the baseline A/B testing method in terms of finding the optimized web form. In addition, Thompson sampling surpassed epsilon-greedy as it focuses more aggressively on exploiting the optimized web form once it becomes certain about the reward probability of the two web forms. We believe our work provides constructive guidance on designing high-quality web forms and also presents a simple yet effective solution for finding the best form variant based on user responses.

8. Future Work

In this study we use Thompson algorithm with Beta distribution where we were restricted to univariate variables, however this can be extended using contextual bandits. Contextual bandits may perform better than Thompsons Sampling as they use information of the current state to choose an action.

As mentioned in section 3.1, the experiment was live for 2 weeks and we were able to reach out to more than 200 people. We believe running simulations with larger data could reveal trends based on the demographics. For example, the optimized form contains 6 employment status options out of which we observe 2 choices in our experiment. With a more diverse participation we could gather more information based on parameters like this.

Larger data will also enable us to implement deep reinforcement learning methods, also called Deep RL. These are sophisticated neural network models that can deal with high dimension unstructured data and select actions based on input, current state and outcome.

9. References

[1] Brown T. Design thinking[J]. Harvard business review, 2008, 86(6): 84.

[2] A. Hanmandla, J. Ranoliya, D. Ojha and S. Kulkarni, "Webform Optimization using Machine Learning," 2021 6th International Conference for Convergence in Technology (I2CT), 2021, pp. 1-5, doi: 10.1109/I2CT51068.2021.9417919.

[3] J. Cruz-Benito, A. Vázquez-Ingelmo, J. C. Sánchez-Prieto, R. Therón, F. J. García-Peñalvo and M. Martín-González, "Enabling Adaptability in Web Forms Based on User Characteristics Detection Through A/B Testing and Machine Learning," in IEEE Access, vol. 6, pp. 2251-2265, 2018, doi: 10.1109/ACCESS.2017.2782678.

[4] Mignon, Alexandre dos Santos and Ricardo Luis de Azevedo da Rocha. “An Adaptive Implementation of ε-Greedy in Reinforcement Learning.” ANT/SEIT (2017).

[5] Guo, D., Ktena, S.I., Huszár, F., Myana, P.K., Shi, W., & Tejani, A. (2020). Deep Bayesian Bandits: Exploring in Online Personalized Recommendations. *Fourteenth ACM Conference on Recommender Systems*.

[6] Li, Yuyao. (2020). Comparison of Various Multi-Armed Bandit Algorithms (Ɛ -greedy, Thompson sampling and UCB-1) to Standard A/B Testing. 10.13140/RG.2.2.31519.69282.

[7] Epsilon Greedy Algorithm: https://medium.com/analytics-vidhya/the-epsilon-greedy-algorithm-for-reinforcement-learning-5fe6f96dc870

[8] Thompson Sampling Algorithm: https://gdmarmerola.github.io/ts-for-bernoulli-bandit/

[9] https://github.com/cbonitz/thompson-sampling-vs-epsilon-greedy

Appendix

The project resources can be found at the following locations:

S. No.	Resource	Location
1	User Interviews	Link
2	Randomized A/B Experiment	Link
3	Data collected from Randomized A/B Experiment	Link
4	Algorithm implementations, results and analysis	GitHub