Webform Optimisation Using Machine Learning

A project report submitted in complete fulfillment of the requirements for the degree

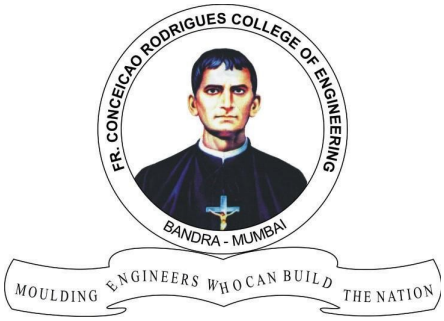
of

Bachelor Of Engineering

by

Akshaykumar Hanmandla (Roll no 7699)
Jaydeep Ranoliya (Roll no 7719)
Dhananjaykumar Ojha (Roll no 7713)

Under the guidance of
Saurabh Kulkarni



DEPARTMENT OF INFORMATION TECHNOLOGY

Fr. Conceicao Rodrigues College of Engineering, Bandra (W), Mumbai - 400050

University of Mumbai
April 25, 2019

This work is dedicated to our family.
We are very thankful for their motivation and support.

Internal Approval Sheet

CERTIFICATE

This is to certify that the project entitled **Webform Optimization using Machine Learning** is a bonafide work of **Akshaykumar Hanmandla(7699), Jaydeep Ranoliya(7719)** and **Dhananjaykumar Ojha(7713)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering in Information Technology**

(Name and sign)
Supervisor/Guide

(Name and sign)
Head of Department

(Name and sign)
Principal

Project Approval Sheet

CERTIFICATE

This project report entitled by **Webform Optimization using Machine Learning** by **Akshaykumar Hanmandla(7699)**, **Jaydeep Ranoliya(7719)** and **Dhananjaykumar Ojha(7713)** is approved for the degree of **Bachelor of Engineering**

Examiners

1._____

2._____

Place :
Date :

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/-data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Akshaykumar Hanmandla (Roll No. 7699) (sign) _____
Jaydeep Ranoliya (Roll No. 7719) (sign) _____
Dhananjaykumar Ojha (Roll No. 7713) (sign) _____

Date :- April 25, 2019

Abstract

Forms are used in websites for improving business and user experience[3]. If the form length is big, then we get more information but information is less accurate and vice versa. Nowadays, A/B testing algorithm is being used for getting the optimal web form from a number of web forms to choose. A/B test distributes the forms uniformly to the visitors. The winning form will be the form which has more no. of conversions. So, after deploying new version of form, A/B test explores all the forms equally[1]. Even though some forms do not perform well, it still explores which is wastage of time and resources. So, there is more time given for exploration in A/B test. And also owner of website during testing period focuses more on testing than running the website, due to which there may be a loss of visitors. To overcome this problem, we can use some automated machine learning algorithms to predict the optimal web form from a number of forms to choose. These algorithms does not waste time and resources exploring inferior forms due to which we will get the results in less time and also we can simultaneously run the website.

Acknowledgements

We have great pleasure in presenting the report on **'Webform Optimization using Machine learning'**. We take this opportunity to express our sincere thanks towards the guide **Mr.Saurabh Kulkarni**, C.R.C.E, Bandra (W), Mumbai, for providing the technical guidelines, and the suggestions regarding the line of this work. We enjoyed discussing the work progress with him during our visits to department.

We thank **Mrs.Garima Tripathi**, Head of Information Technology Dept., Principal and the management of C.R.C.E., Mumbai for encouragement and providing necessary infrastructure for pursuing the project.

We also thank all non-teaching staff for their valuable support, to complete our project.

Akshaykumar Hanmandla (Roll No. 7699) (sign) _____
Jaydeep Ranoliya (Roll No. 7719) (sign) _____
Dhananjaykumar Ojha (Roll No. 7713) (sign) _____

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

The use of websites is increasing day by day. By considering the past trends, the use of websites has increased exponentially from the last decade. Websites are used for running the businesses online, forums, social media sites, etc. Forms are used in websites for improving business and user experience as well as for various purposes. The owners of website get feedback from the form and accordingly they can adjust their services. Most of the users coming to the websites don't fill the form due to lack of interest and owner of website don't get appropriate feedback. There are many solutions to overcome this problem. The most familiar solution is A/B testing.

1.2 Motivation

Nowadays, there are more websites coming into the online market. Forms play a major role in improving conversions of a websites. There may be a situation that user may not like to fill the form which causes loss of conversions[5]. To overcome this problem, we have to choose the best form so that most users will fill the form. In choosing the best version of a form on a page, most people use A/B testing. But it discretely jumps from exploration to exploitation. And, it gives equal chance to the inferior forms which do not perform well which leads into wastage of time and resources.[4] This is the area our project wants to work into: selecting the best form to put into a website so that more users will fill. And to develop an algorithm gives the best form in less time and efficient way than A/B testing.

1.3 Overview

In this experiment, we are comparing two algorithms for selection of best form out of no. of forms to choose.

1.3.1 A/B test

This algorithm distributes the available forms uniformly to the visitors visiting the website. This method has been found effective in various testing environments. Most of the people use this type of test to test whether the change in any website would be good or bad. The same technique we are applying for selecting the best form[7].

1.3.2 Adaptive Epsilon Greedy

An initial epsilon value has to be set. As the algorithm starts running, the epsilon value adjusts automatically (increases or decreases) according to the responses given by the user. Fully automated testing. Algorithms behaves according to the previous experiences learnt while in A/B test behaves the same in all the conditions[7].
At the end, we compare the performance of both the algorithms.

1.4 Scope

We have implemented an algorithm which selects the best and optimal form in an efficient way. It learns by itself and It explores as well as exploits (Earn while you learn). We have experimented for three forms and it can be done for n number of forms. It shows best possible version while it exploits rather than showing bad version which does not perform well. It can be implemented for other elements of websites like logo, banner, background colour, navigation bar, etc. This algorithm can be extended for entire web page or website also.

Chapter 2

Literature Review

2.1 Enabling Adaptability in Web Forms Based on User Characteristics Detection Through A/B Testing and Machine Learning

This paper presents an original study with the aim of improving users' performance in completing large questionnaires through adaptability in web forms. Such adaptability is based on the application of machine-learning procedures and an A/B testing approach. To detect the user preferences, behavior, and the optimal version of the forms for all kinds of users, researchers built predictive models using machine-learning algorithms (trained with data from more than 3000 users who participated previously in the questionnaires), extracting the most relevant factors that describe the models, and clustering the users based on their similar characteristics and these factors. Based on these groups and their performance in the system, the researchers generated heuristic rules between the different versions of the web forms to guide users to the most adequate version (modifying the user interface and user experience) for them. To validate the approach and confirm the improvements, the authors tested these redirection rules on a group of more than 1000 users. The results with this cohort of users were better than those achieved without redirection rules at the initial stage. Besides these promising results, the paper proposes a future study that would enhance the process (or automate it) as well as push its application to other fields [1].

2.2 Online Controlled Experiments and A/B Testing

The Internet connectivity of client software (e.g., apps running on phones and PCs), websites, and online services provide an unprecedented opportunity to evaluate ideas quickly using controlled experiments, also called A/B tests, split tests, randomized experiments, control/treatment tests, and online field experiments. Unlike most data mining techniques for finding correlational patterns, controlled experiments allow establishing a causal relationship with high probability. Experimenters can utilize the scientific method to form a hypothesis of the form "If a specific change is introduced, will it improve key metrics?" and evaluate it with real users. The theory of a controlled experiment dates back to Sir Ronald A. Fisher's experiments at the Rothamsted Agricultural Experimental Station in England in the 1920s, and the topic of offline experiments is well developed in Statistics (Box et al., Statistics for experimenters: design, innovation, and discovery. Wiley, Hoboken, 2005). Online-controlled experiments started to be used in the late 1990s with the growth of the Internet. Today, many large sites, including Amazon, Bing, Facebook, Google, LinkedIn, and Yahoo!, run thousands to tens of thousands of experiments each year testing user interface (UI) changes, enhancements to algorithms (search, ads, personalization, recommendation, etc.), changes to apps, content management system, etc. Online-controlled experiments are now considered an indispensable tool, and their use is growing for startups and smaller websites. Controlled experiments are especially useful in combination with Agile software development (Martin, Clean code: a handbook of Agile software craftsmanship. Prentice Hall, Upper Saddle River, 2008; Rubin, Essential scrum: a practical guide to the most popular Agile process. Addison-Wesley Professional, Upper Saddle River, 2012), Steve Blank's Customer Development process (Blank, The four steps to the epiphany: successful strategies for products that win. Cafepress.com., 2005), and MVPs (minimum viable products) popularized by Eric Ries's Lean Startup (Ries, The lean startup: how today's entrepreneurs use continuous innovation to create radically successful businesses. Crown Business, New York, 2011)[2].

Chapter 3

Problem Description

3.1 Problem Statement

Forms are used in websites for signup and getting feedback from users. Sometimes, there may be a problem if owner wants to add new fields to the form because customers may feel difficult which in turn affect business.To develop a system which will help the business to not go into loss while testing the website forms. Owners can earn while learning. The strategy used in the website for testing the users will not affect the ROI (Return On Investment) as in A/B testing[6]. Implicit indirect feedback will be taken from visitors (without knowing them) for new web forms. Gives results to owner regarding changes in a particular website form will actually attract the visitors or reduce the number of visitors. So, we implemented an algorithm which gives an optimal web form from a no. of web forms to choose.

3.2 Data flow diagram

A data flow diagram gives the basic idea of the flow of data in an information system while giving a short overview of its processing steps.

3.2.1 DFD Level 0

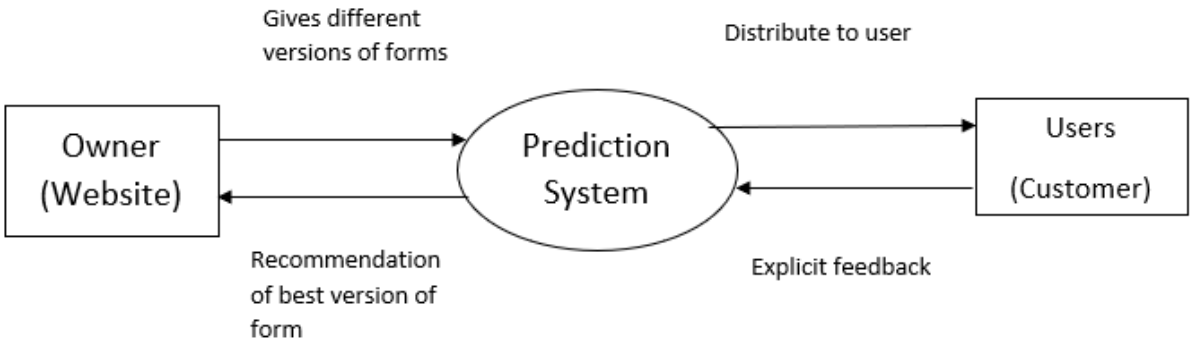


Figure 3.1: DFD level 0

At DFD Level 0 there are 2 entities which are, Website and the User. The website provides the form which user fills. The filled information is sent to the Prediction System which gives optimal of the form for a website.

3.2.2 DFD Level 1

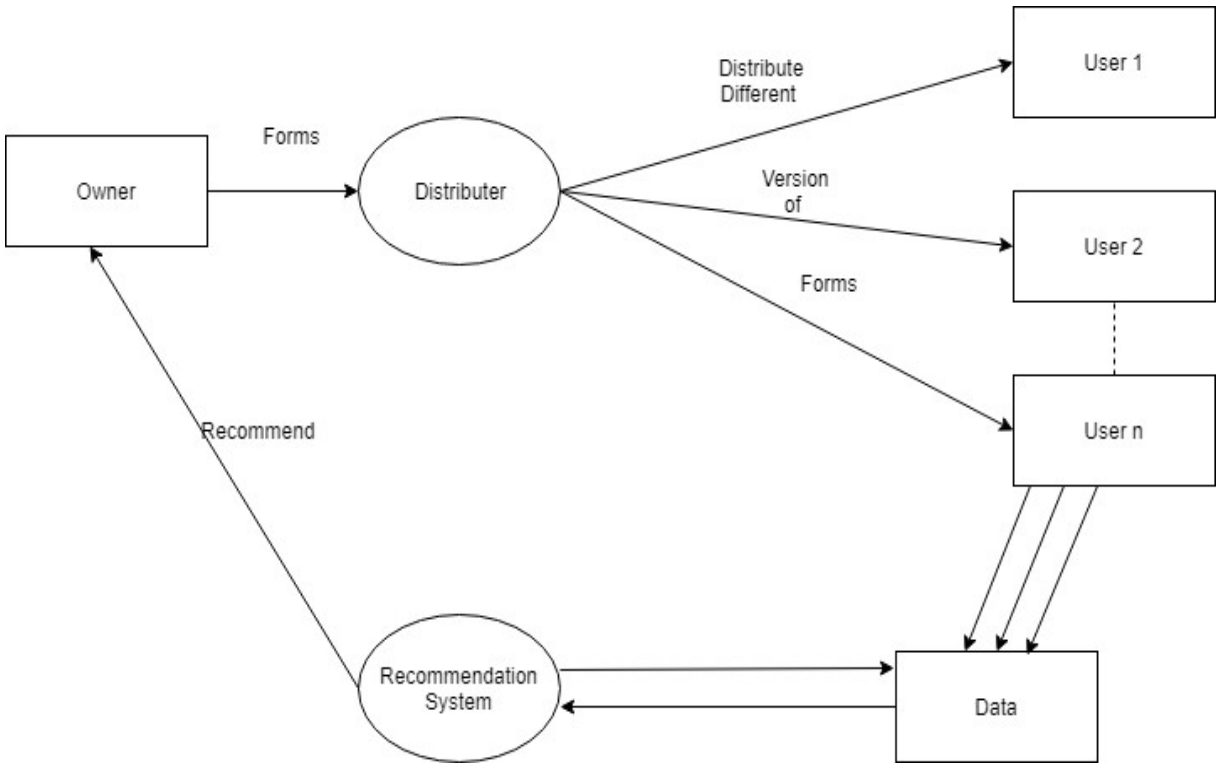


Figure 3.2: DFD level 1

At DFD Level 1, the website gives the forms to the algorithm. The algorithm distributes the forms to the users based on some probability.Then, user fills the form. The user's characteristics of filling the form is stored in the database. The prediction system uses the data in the database and provides the optimal webform to the website owner.

3.3 Design

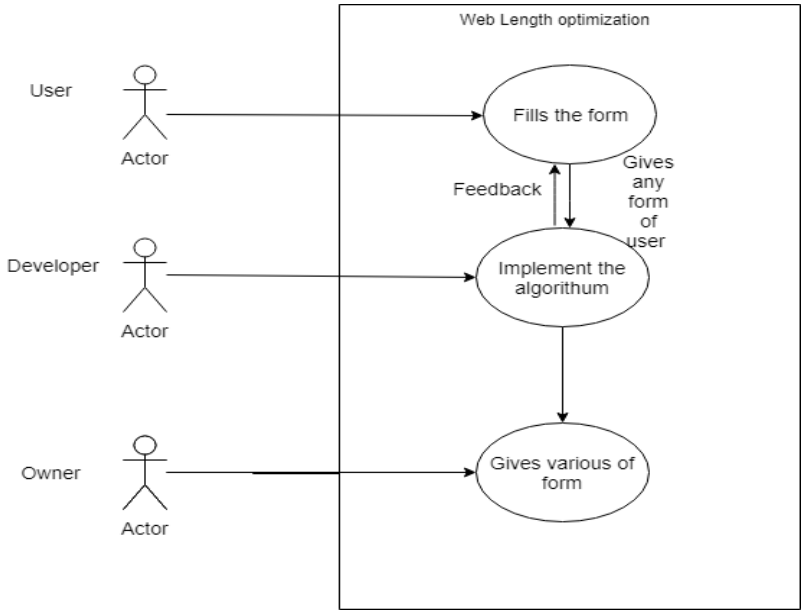


Figure 3.3: Usecase Diagram

In the above use case diagram, there are three actors 1) User 2) Developer 3) Owner. Owner gives various forms. Developer wirtes the algorithm and integrates it with the form.User fills the form from which input is taken.

Chapter 4

Methodologies and Technologies

4.1 Metholodology

1. Choose the new versions of form.
2. Determining performance parameters of form.
3. Integrating all the forms with A/B test algorithm.
4. Implement A/B test. Test the forms with real users which stores their filling characteristics in PostgreSQL DB.
5. Integrating all the forms with Adaptive Epsilon Greedy algorithm.
6. Implement Adaptive epsilon greedy test. Test the forms with real users which stores their fiiling characteristics in PostgreSQL DB.
7. Compare the performance of both the algorithms through the results stored in database.
8. Plotting the graph which shows the learning rate of both the algorithms.

4.2 Performance Parameters

A form's performance is measured by two conditions. A scored is assigned for every after user fills the form. Based on user filling the form, the score is assigned. The conditions for assigning score is as follows:-

1. A form will be assigned score 1 if the user fills 75% of the total fields in the form.
2. A form will be assigned score 1 if the user fills the form in more than 15 seconds.
3. If both conditions 1 and 2 passes, then form will be assigned score 2.

4.3 Technology and Algorithm

4.3.1 Technology

The main programming language which will be used is Python Django.

Software : Python IDEs, MS Excel, PostgreSQL, Atom editor, Git, Github, Django.

Hardware : Intel Core 2 or AMD Athlon 64 processor; 2 GHz or faster processor. Microsoft Windows 7, Windows 8.1, or Windows 10. Any Linux based OS can work too. 2 GB or more of RAM (8 GB recommended)

4.3.2 Algorithm

A/B test algorithm

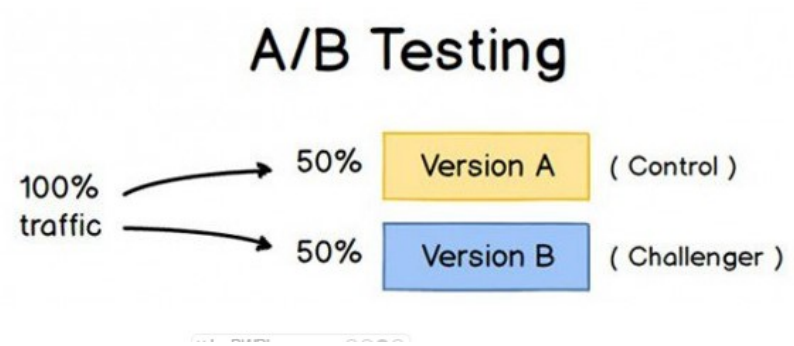


Figure 4.1: Process of A/B test Algorithm

Steps in A/B testing algorithm:-

1. For every user visiting the form, a form out of all the forms to choose is shown randomly.
2. When user submits a particular form, corresponding form score is updated.
3. When new user comes, go to step 1.
4. At the end of the testing period, each form's performance is compared and best form is decided.

Adaptive Epsilon Greedy test Algorithm

1. Set explore min count=0, X=initial epsilon value, N= no. of people to test, T=0 (no. of people submitted the form at a situation)
2. For every user visiting the form, check if explore min count equals M. If yes, set explore min count equals 0 and go to step 3 else go to step 4.
3. Calculate new epsilon value.
4. Select epsilon value from database and check for exploration or exploitation. Go to step 5.
5. If its explore, then show random form to user else show best form to user.
6. If user submits the form, then collect user characteristics and store details in the database. Update the performance of form and statistics in the database.
7. If T== N, then go to step 8, else go to step 2.
8. Stop running the algorithm and always show the best form which is decided by the performance in the test.

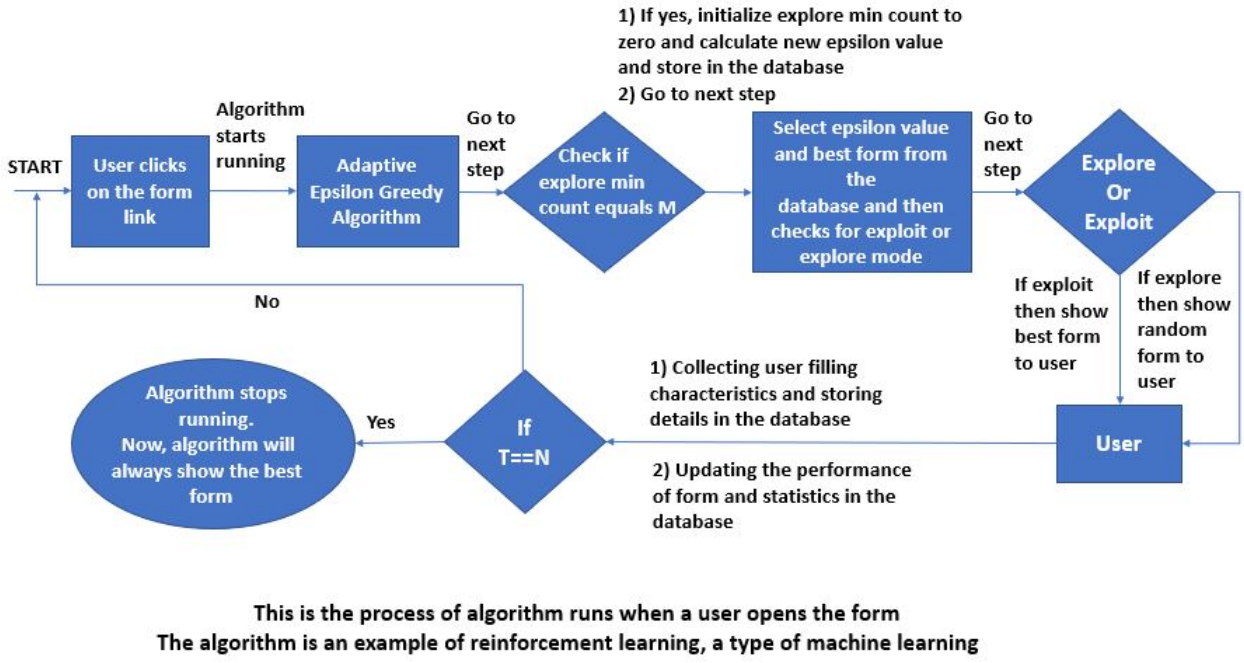


Figure 4.2: Process of Adaptive Epsilon Greedy Algorithm

Explore min count (between 0 and M) = The count which tells the no. of explorations done and initialized to 0 after M explorations.
M = The no. of explorations after which the best form has to be selected and epsilon value to be adjusted.
T = The user no. of last user who filled the form.
N = The no. of users for which the test has to be conducted.

Chapter 5

Implementation detials

The different versions of form has to integrated with Adaptive Epsilon Greedy Algorithm. An initial epsilon value has to be set. As the algorithm starts running, the epsilon value adjusts automatically (increases or decreases) according to the responses given by the user. For eg:- If the epsilon value is 0.1, then exploration will be done 10% of the time and exploitation will be done 90% of the time. And suppose, if the performance in the exploration phase will be good, then epsilon value increases depending on the percentage increase or decrease in the performance of the exploration phase. The epsilon value cannot become greater than 0.5.The test should be done for the number of users specified by the website owner. However, the more the versions of forms, the more the users should be considered. The more the number of users, the accurate will be the results.
After the end of experiment, the best form will be the one which has maximum score. The owner can also see the performance of the form on the dashboard. The performance of forms is shown in the form of curve. If the owner of website is not satisfied with the results, he can still continue the test with some set of users.

The questions asked in form are based on personal information. Question asked in the form are given below.

1. What is your first name?
2. What is your middle name?
3. What is your surname?
4. What is your age category?
5. What are your hobbies?
6. Have you participated in any social service camp? If yes, what is it?
7. What is your favourite sport?

8. What is your favourite TV show? Which food item do you like?

There were three forms called form 1, form 2 and form 3.
Form 1 contains question numbers 1 2 3 4 5 6 7 8 9
Form 2 contains question numbers 1 2 3 4 5 6 9
Form 3 contains question numbers 1 2 3 4 5 6

5.1 A/B test table fields

The table for A/B tests stores the following information apart from the fields

A/B test table details	
Column	**Definition**
user no	The serial no. of user who fills the form. It has autoincrement property.
version no	The version of form.
hash	Hash value is used to recognize the user properly at the server side.
Filled	If the user submits the form, then it contains "yes" else "no".
Score	Score assigned to form out of 2.
no_filled	The no of fields filled by the user
no_of_Fields	The no. of fields in the form.
created at	It contains date time value at which the record is inserted.
Percentage	Percentage of number of fields filled by the user out of total fields
fill time	Time taken to fill the form.

5.2 Adaptive Epsilon Greedy tables

5.2.1 Epsilon Greedy form fields

There are two tables for Epsilon Greedy Algorithm.
The first table stores the user filled info.

Epsilon Greedy test table details	
Column	**Definition**
user no	The serial no. of user who fills the form. It has autoincrement property.
Experiment	The experiment which is being performed (Explore/Exploit)
version no	The version of form. It can be 1, 2 or 3.
hash	Hash value is used to recognize the user properly at the server side.
Filled	If the user submits the form, then it contains "yes" else "no".
Score	Score assigned to form out of 2.
no_filled	The no of fields filled by the user
no_of_Fields	The no. of fields in the form.
created at	It contains date time value at which the record is inserted.
Percentage	Percentage of number of fields filled by the user out of total fields
fill time	Time taken to fill the form.

5.2.2 Performance fields of Epsilon Greedy

The second table contains the performance parameters of each form and also the fields used to calculate the performance paramters.

Epsilon Greedy test performance table details	
Column	**Definition**
Exploit	The no. of times exploitation done till the current user.
Explore	The no. of times exploration done till the current user.
Total	The sum of exploration and exploitation.
M	It tells how many times to perform the exploration mode before the adaptive action which changes the value of Epsilon.
Explore min count	It tells how many times exploration mode has been performed after the adaptive action which changes the value of Epsilon.
Previous explore percentage	The exploration percentage at which Explore min count becomes 0.
Current explore percentage	The current exploration percentage. When explore min count equals 0, then current explore percentage is assigned to previous explore percentage.
Epsilon	The value of epsilon which decides the percentage of exploration and exploitation to be done.
Form 1	It gives the average score of form 1 (form1 score/form1 count) .
Form 2	It gives the average score of form 2 (form2 score/form2 count)
Form 3	It gives the average score of form 3 (form3 score/form3 count)
X	The initial epsilon value.
Form 1 score	The sum of all the scores earned by form 1 till the current user.
Form 2 score	The sum of all the scores earned by form 2 till the current user.
Form 3 score	The sum of all the scores earned by form 3 till the current user.
Best	The best form at the current situation.
Explore score	The sum of all the scores scored in the exploration mode.)

Chapter 6

Results

- We made a dashboard for admin to check performance of each form. The results consists of exploration percentages of A/B test and Adaptive Epsilon Greedy test as well as the performance of each form of both the tests.
- For the performance graph of both algorithms,
- The X-axis defines the no. of users and Y- axis defines the formula of each form.
- The formula is given as
- Performance of form i after n users filling the form (i=form number) is given as Performance = (Cumulative score of a form i till the nth user)/(2*n)

6.1 A/B test

6.1.1 Exploration percentage

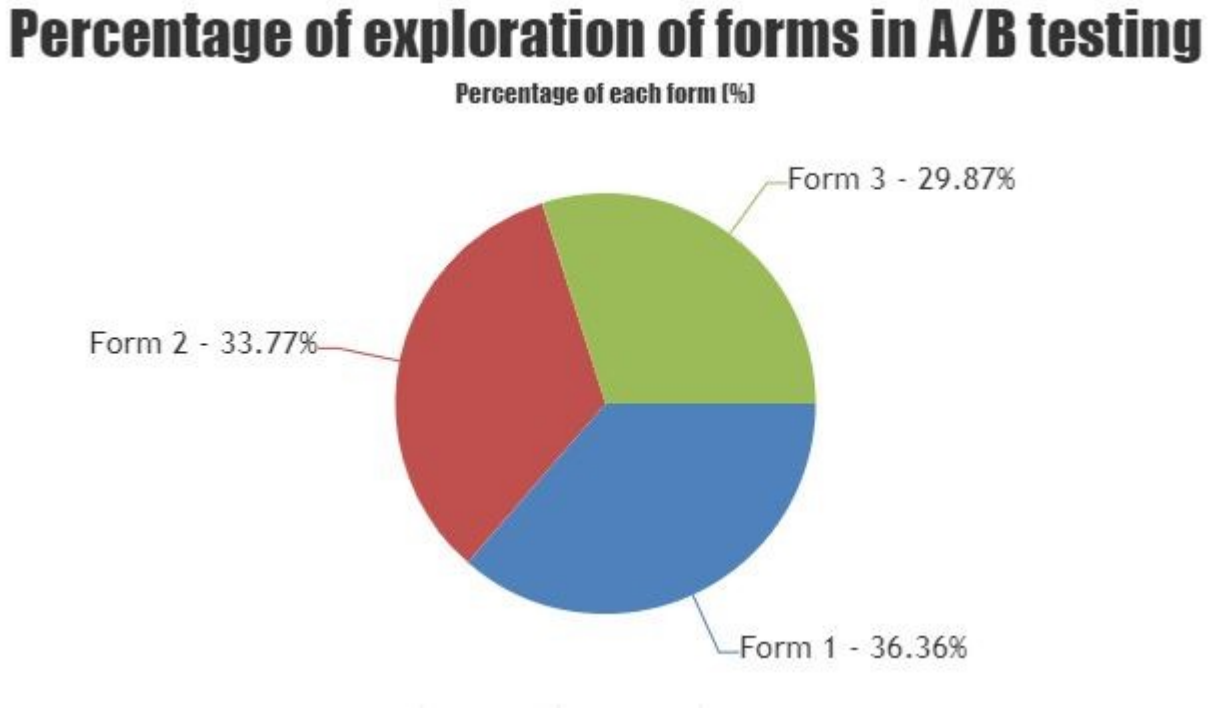


Figure 6.1: Exploration percentage of each form in A/B testing

- In the case of A/B testing, each form gets approximately equal chances for exploration.
- As per the above Pie chart, Form 1, Form 2 and Form 3 are explored for 36.36%, 33.77%, 29.87% respectfully.
- So, by seeing the percentages, it seems that each form is approximately equally explored.

6.1.2 Performance of each form

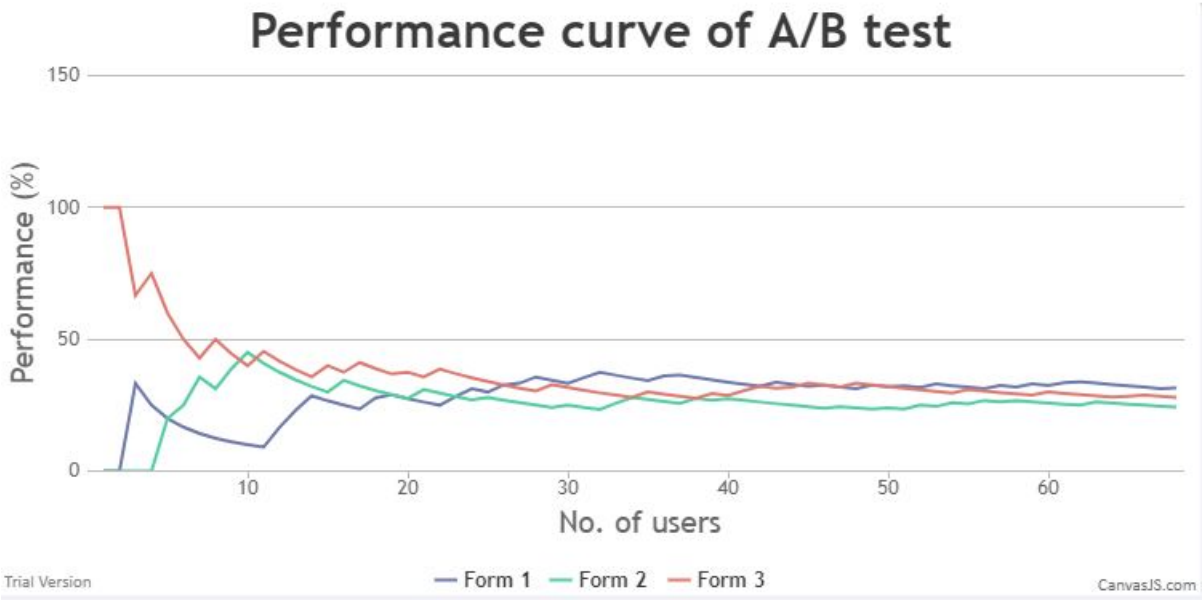


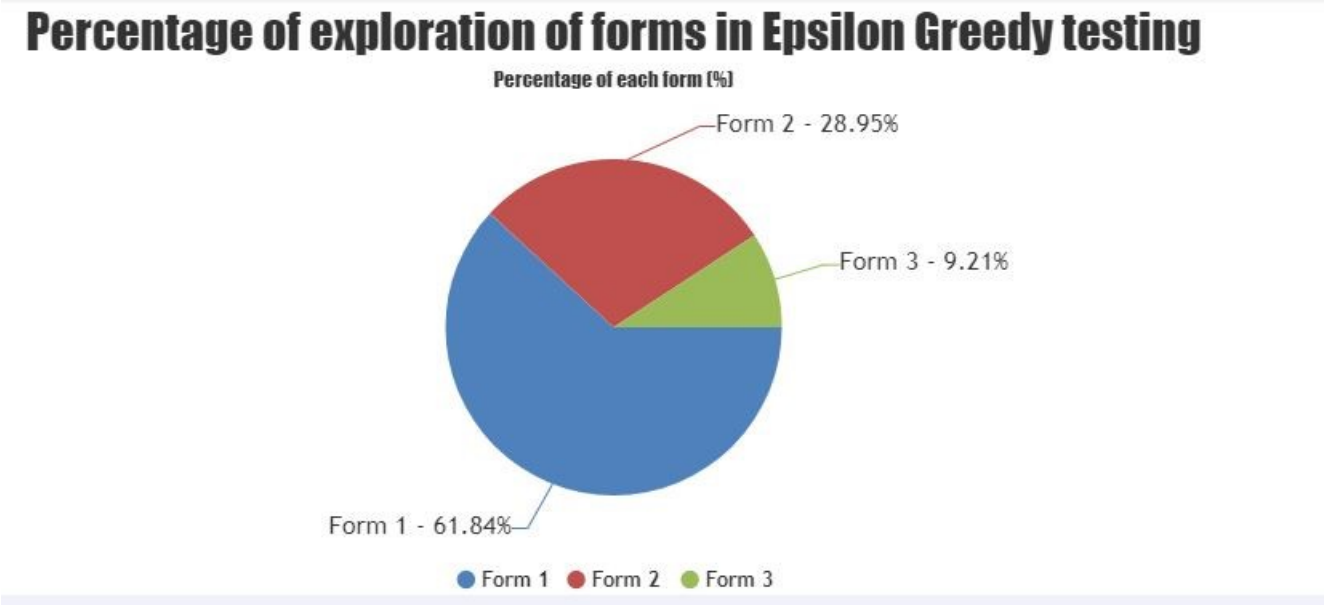
Figure 6.2: Performance of each form in A/B testing

- From the above graph, we can see that performance of form 1 is better than the other two forms and it has become constant later. So, we can conclude that form 1 is the best form.
- As we see in performance chart, form 2 and form 3 was not performing well, we still explored for 33.77% and 28.87% respectfully as given in the pie chart
- So, during exploration using A/B testing we didn't earn anything and we lose conversions.

6.2 Adaptive Epsilon Greedy Test

6.2.1 Exploration percentage

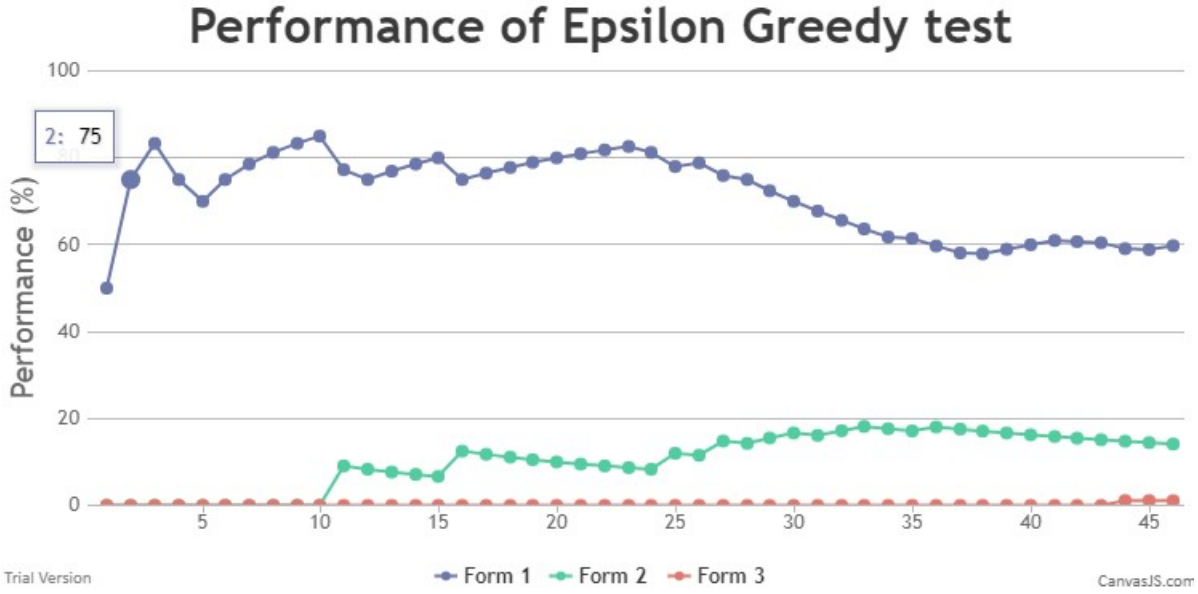
Figure 6.3: Exploration percentage of each form in Adaptive Epsilon Greedy Testing



- In the case of the Adaptive Epsilon Greedy Algorithm, exploration of the forms is done as per epsilon value.
- As per the above Pie chart, Form 1, Form 2 and Form 3 are explored for 61.84%, 28.95%, 9.21% respectfully.

6.2.2 Performance of each form

Figure 6.4: Performance of each form in Adaptive Epsilon Greedy Testing



- From the above graph, we can see that performance of form 1 is better than other two forms.
- Hence, form 2 and form 3 has been explored less due to their poor performance which algorithm came to know at the beginning.
- So, exploration for Form 2 and Form 3 is reduced to 28.95% and 9.21% respectfully.
- But still it gave chance to form 2 and form 3 sometimes by expecting that it may perform well in the future. The algorithm's behaviour changes according to the situation. This does not happen in A/B test. A/B test behaviour is same every time no matter which version performs well.
- Hence, we are not losing conversions in Adaptive Epsilon Greedy as we lose in A/B test.

Chapter 7

Conclusion

7.1 Applications

An organization can predict the future performance of the form without losing customers. An organization can increase their revenue. This algorithm can be implemented in websites having sign up forms, feedback forms and also social media sites.This project helps organization and developer to make web form very efficient and to get needed information from users by taking implicit feedback.

7.2 Future enhancements

We have implemented our algorithm for Web form. It can be implemented for other elements of website like logo, background color, navigation bar etc. This algorithm can be extended for entire web page or web site.

7.3 Final conclusion

Using the modern technology web stack, we have built a system which will be able to find the best form in an efficient way. We have performed A/B test and Adaptive epsilon greedy test for forms of three versions with a set of users and found out that Adaptive epsilon greedy algorithm finds the best form quickly and efficiently than A/B test. The fields we used in the form were based on personal information.This algorithm will help us to save lot of time and resources while exploring. Thus project on web form optimization using machine learning has been successfully implemented.

Bibliography

[1] Juan Cruz-Benito, Andrea Vázquez-Ingelmo, José Carlos Sánchez-Prieto, Roberto Therón, Francisco José García-Peñalvo, and Martín Martín-González. Enabling adaptability in web forms based on user characteristics detection through a/b testing and machine learning. *IEEE Access*, 6:2251–2265, 2018.

[2] Ron Kohavi and Roger Longbotham. Online controlled experiments and a/b testing. In *Encyclopedia of machine learning and data mining*, pages 922–929. Springer, 2017.

[3] Steve Krug. *Don't make me think!: a common sense approach to Web usability*. Pearson Education India, 2000.

[4] Tor Lattimore and Csaba Szepesvári. Bandit algorithms. *preprint*, 2018.

[5] Jakob Nielsen. *Usability engineering*. Elsevier, 1994.

[6] Mirjam Seckler, Silvia Heinz, Seamus Forde, Alexandre N Tuch, and Klaus Opwis. Trust and distrust on the web: User experiences and website characteristics. *Computers in Human Behavior*, 45:39–50, 2015.

[7] John White. *Bandit algorithms for website optimization*. " O'Reilly Media, Inc.", 2012.