Cover Page – MSc Business Analytics Consultancy Project/Dissertation 2023-24

Candidate #: JHSM3

Title of Project: Hypothesis Testing Optimization: Development of Variance Reduction Techniques in Mobile Gaming Data

Date: 4th Aug 2024

Word Count: 11,664

Disclaimer:

I hereby declare that this dissertation is my individual work and to the best of my knowledge and confidence, it has not already been accepted in substance for the award of any other degree and is not concurrently submitted in candidature for any degree. It is the end product of my own independent study except where other acknowledgement has been stated in the text.

Marking Sheet – MSc Business Analytics Consultancy Project/Dissertation 2023-24

Criteria/Weight	Supervisor's comments
Topic, theoretical framework, literature, and methodology (35%): Topic is clearly identified and boundaries are asserted. Knowledge of relevant theories and their limitations. Current and relevant literature coming from reliable sources. Appropriate and adequate methodology for topics. Detailed methodology facilitating replication of project and reproducibility of results.	
Analysis and conclusions /recommendations (35%): Use of primary and/or secondary data. Rigorous analysis and interpretations. Alternative interpretations/arguments are considered. Limitations are identified and justified by reasonable arguments. Conclusions/recommendations are fully consistent with evidence presented.	
Structure, originality and presentation (10%): Provides a concise summary. Demonstrates an understanding of business context. Coherent and appropriate structure. Adequate presentation, language, style, graphs, tables, and referencing. Appropriate use of visualisation. Presents business recommendations.	
Complexity of project scope and progress made towards business goals (10%): Progress made towards overcoming technical and operational challenges encountered during the project. Progress made in overcoming problem framing and theoretical and data related problems encountered during the project.	
Project Management (10%): Good use of project management and communication tools. Use of Kanban board for structuring project work. Evidence of objectives being broken down in appropriate tasks and timely engagement with primary supervisor.	

General marking guidelines

85+ Outstanding work of publishable standard.

70-84 Excellent work showing mastery of the subject matter and excellent analytical skills.

60-69 Very good work. Interesting analysis with original insights. Some minor errors.

50-59 Good work which only covers a basic analysis. Some problems but no major omissions.

40-49 Inadequate work. Not sufficiently analytical. Some major omissions.

39- Work seriously flawed. Lack of clarity and argumentation. Too descriptive.

Mark: _____

London's Global University

Hypothesis Testing Optimization: Development of Variance Reduction Techniques in Mobile Gaming Data

University College London
Faculty of Engineering
UCL School of Management

A Thesis Presented for the degree of
Master's in Business Analytics
London, United Kingdom
August 2024


tripledot

Abstract

The project aimed to develop and evaluate variance reduction techniques for hypothesis tests for the mobile game company Tripledot to help the company make data-driven decisions easily. We built seven models, including the baseline, based on three techniques: post-stratification, CUPED, and CUPAC. We evaluated the developed models in two tests: an A/A test and implementation on actual A/B tests. In the simulated A/A test, we found that Model 7 (CUPAC with feature selection), Model 6 (CUPAC with pre-experiment data), Model 4 (CUPED with ML), and Model 3 (Basic CUPED) reduced 68% to 60% of variance compared to the baseline. Then, those four models were implemented on two actual A/B tests, one each of a new-user test and an all-user test conducted recently in the company. The results showed that Model 7 (CUPAC with feature selection) performed the best in both A/B tests; Model 4 (CUPED with ML) reduced many variances but was not reliable due to bias; Model 3 (Basic CUPED) performed well in the all-user test but poorly in the new-user test. Finally, due to the business aspect, Model 7 (CUPAC with feature selection) and Model 3 (Basic CUPED) were recommended to the company. The study provides insights into variance reduction models suitable for the mobile gaming industry and suggests further trials to enhance CUPAC's capabilities.

Keywords:

A/B testing, variance reduction techniques, post-stratification, CUPED, CUPAC, machine learning, casual games, mobile game, hypothesis testing

Table of Contents

Lists of Tables

Lists of Figures

Introduction

1.1 Company Overview

Tripledot is a UK-based mobile game studio with offices in six cities worldwide, providing entertaining games to millions of players daily. The company aims to create successful and enjoyable casual games for the public. By combining creativity and data, Tripledot continues to grow rapidly and was recognized as one of the fastest-growing companies in Europe by the Financial Times in 2023 (Tripledot Studios, 2024; Kilby, 2023).

Tripledot has 46 games, including 21 Android and 25 iOS apps. Woodoku, Triple Tile, Tile Dynasty, and Solitaire are the most successful games developed by Tripledot, achieving competitive numbers of new installs and ranking high on the App Store.

1.2 Problem Statement and Research Gap

Tripledot is a data-driven company that relies heavily on A/B tests for product changes and strategies, making these tests critical for growth. Therefore, implementing a robust hypothesis testing structure is crucial. The data science team at Tripledot seeks to enhance the A/B test environment to capture more detailed effects. We have focused on developing variance reduction techniques to increase statistical power, enabling more accessible and more confident decision-making.

Despite the widespread use of variance reduction techniques in e-commerce and streaming, there is a noticeable gap in practical cases within the mobile game industry. Additionally, implementation varies across companies due to differences in data and business needs, highlighting the necessity for self-research and tailored variance-reduction techniques.

1.3 Purpose

The main objective of this dissertation is to develop a customized variance reduction technique for Tripledot, enabling the company to make decisions more efficiently based on A/B test results. To achieve this goal, several sub-goals have been defined:

1. Develop and implement various statistical models based on academic research and theoretical foundations, as no existing code packages are suitable for practical use.
2. Evaluate the performance of each developed model.
3. Provide a recommendation based on the evaluation results to determine the most effective variance reduction technique for Tripledot.

1.4 Structure

The dissertation comprises six chapters. The first chapter briefly introduces the background and the aim of the project. The second chapter, the literature review, explains statistical theories and related academic research, providing a trustworthy foundation for developing our variance reduction technique. The third chapter, the methodology section, describes the experimental process and the evaluation standards used in the study. In the fourth chapter, we carefully demonstrate the design of each model and present the results from running these models in simulated tests. The fifth chapter focuses on selecting potential models, developing them more practically, and applying them to actual A/B test data to evaluate their actual performance. Finally, the sixth chapter concludes with recommendations based on the findings from the practical implementation of the models.

Literature Review

This chapter introduces the concept of online A/B testing and elaborates on potential issues that may arise during an online experiment. After identifying the challenges related to detecting minor effects, multiple methods to address this problem will be introduced and explained.

2.1 Overview of A/B test

Online controlled experiments, or A/B testing, are among the most effective ways to establish causality statistically (Keppel et al., 1998). This methodology allows companies to make rapid, data-driven decisions and evaluate the impact of implementing changes. A/B testing is widely used in the tech industry; companies like Airbnb, Facebook, and Amazon conduct about ten thousand A/B tests annually to improve their online products (Gupta et al., 2019). This extensive use demonstrates the heavy reliance businesses have on A/B testing. A/B tests can be applied to various features, such as user interface and algorithms, to detect differences in adjusted features that might significantly impact the entire company.

A/B testing is to compare the target metrics of different product versions to determine whether the changes influence user behaviour. Users are allocated to at least two groups: control and treatment. The control group uses the original version of the product, while the treatment groups, which can be multiple, use slightly different versions. The allocation is based on randomization, which minimizes the impact of external features on the target metrics.

Hypothesis testing provides the statistical foundation for A/B testing, commonly using a two-sample, two-tailed t-test. Considering their variance, this test measures the difference in means between the control and treatment groups. The null hypothesis usually posits that "the control group and the variant have the same effect on users," indicating that the target metrics are identical for both groups (see equation 1):

$$H_0 : mean\ of\ Y^{Treatment} = mean\ of\ Y^{Control}$$
$$H_1 : mean\ of\ Y^{Treatment} \neq mean\ of\ Y^{Control} \quad (1)$$

The significance level is set before the experiment and defines the probability of rejecting the null hypothesis when it is true. The p-value indicates the probability of obtaining results as extreme as the sample results, assuming the null hypothesis is correct. The null hypothesis is rejected when the p-value is less than the significance level, suggesting that the result observed in the variant is too extreme, assuming it is the same as the control group. For instance, in a two-tailed test with a significance level of 0.05, there is a 5% risk of incorrectly concluding a difference between the groups. Statistical significance is determined when the p-value in either tail (upper or lower) is less than 0.025 Another method to evaluate A/B tests is to check if the confidence interval for the difference in means includes zero. A 95% confidence interval implies that the actual difference between the two groups will fall within this range 95% of the time across repeated tests.

2.2 Overview of Variance Reduction Techniques

Enhancing an experiment's statistical power, which is the probability of correctly detecting the treatment effect, is crucial for accurate analysis. Statistical power is inversely related to Type II error (β). In hypothesis testing, Type I errors (α) occur when we wrongly reject a true null hypothesis, while Type II errors happen when we fail to reject a false null hypothesis, as shown in Table 1. Statistical power, calculated as (1 - β), is the probability of avoiding a Type II error, ensuring correct rejection of a false null hypothesis, thus yielding an accurate result (Corotto, 2022).

Table 1

Potential Outcomes of Hypothesis Testing

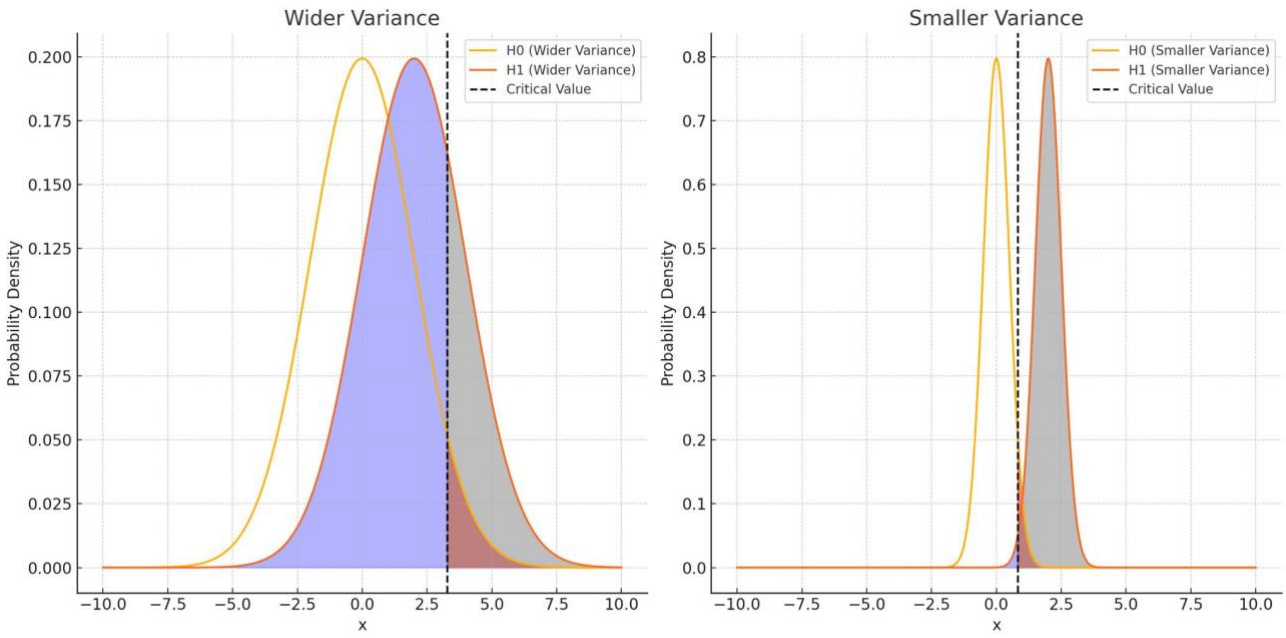
H_0 is...	True	False
Not rejected	Correct decision Probability = 1 - α	Type II error Probability = β
Rejected	Type I error Probability = α (significance level)	Correct decision Probability = 1 – β (statistical power)

Note. Adapted from Bhandari, P. (2021, Jan 18). *Type I & Type II Errors | Differences, examples, visualizations.* Scribbr. https://www.scribbr.co.uk/stats/type-i-and-type-ii-error/

An experiment with higher sensitivity results in greater statistical power. Sensitivity refers to the ability to discern whether the experimental result is due to the treatment effect or a sampling error (Murphy & Myors, 2023). Low sensitivity can lead to unreliable results, such as observing a significant treatment effect when there is none, with differences arising purely from sampling error. Increasing sample size is the most intuitive method to improve sensitivity, though it may not always be practical in industry settings. Alternatively, reducing sample variance can also enhance sensitivity. Figure 1 shows that smaller variance leads to higher sensitivity and statistical power. The orange area represents Type I error probability, the purple area represents Type II error probability, and the grey area represents statistical power (1 - β). The grey area in the graph with smaller variance is larger than in the graph with broader variance, indicating that tests with smaller sample variance have higher statistical power and better ability to draw correct conclusions.

Figure 1

Impact of Variance on Hypothesis Testing Distributions



Note. The only difference between the two graphs is the sample variance. Both graphs have the same significance value of 0.05 and the same target metric (x) for both H_1 and H_0 distributions. The mean of H_0 is 0, and the mean of H_1 is 2. The standard deviation of the left plot is 2, and the standard deviation of the left plot is 0.5.

Several methods can reduce variance, such as transforming the metric to binarization or using a logarithmic scale. However, these transformations can make metrics difficult to interpret and potentially lead to misunderstandings. Alternatively, techniques like stratification and covariate control tools such as CUPED (Controlled Experiments Using Pre-Experiment Data) (Kohavi et al., 2020; Deng et al., 2013) are widely used. These methods reduce variance without compromising metric interpretability.

2.3 Stratification and Post-Stratification

Stratified sampling is a standard method to reduce variance before sampling. This involves dividing the population into distinct subgroups, known as strata. Experimental units are then sampled

independently and randomly from each stratum, ensuring that each subgroup is adequately represented (Cochran, 1977). According to Xie and Aurisset (2016), the variance when using stratified sampling is smaller than when using simple random sampling because stratification removes the within-strata variance. The variance of the estimate using stratification is given by Equation 2, while the variance using simple random sampling is given by Equation 3. The variance difference between the two methods is $\frac{1}{n}\sum_{k=1}^{K} p_k(\mu_k - \mu)^2$.

The notations were defined as follow:

- p_k is the proportion of the population of stratum K
- μ_k is the mean of the target metric in stratum K
- μ is the population mean of the target metric
- n is the number of users in a variant from all stratums
- σ^2 is the population variance of the target metric
- ${\sigma_k}^2$ is the K stratum variance of the target metric

$$Var(Y_{strat}) = \frac{1}{n}\sum_{k=1}^{K} p_k {\sigma_k}^2 \tag{2}$$

$$Var(\bar{Y}) = \frac{1}{n}\sigma^2 = \frac{1}{n}\sum_{k=1}^{K} p_k {\sigma_k}^2 + \frac{1}{n}\sum_{k=1}^{K} p_k(\mu_k - \mu)^2 \tag{3}$$

However, implementing stratification can be costly and time-consuming, requiring dividing the population into strata before assignment. These pre-assignment requirements are often impractical in many real-world scenarios.

Post-stratification offers an alternative approach to achieving effectiveness similar to stratification. This method involves placing observations into strata after conducting simple random sampling and then adding weights to ensure the sample reflects the distribution of characteristics in

the population. This adjustment ensures that the sample is more representative. For example, if a stratum is underrepresented in the sample compared to the population, it receives higher weights to balance the distribution (Holt & Smith, 1979; Glasgow, 2005).

To implement post-stratification in an A/B test, rules to allocate individuals into strata and assignment to treatment or control groups must be defined. Therefore, each row will be assigned to one stratum and one variant. Then, the weight to balance the sampling must be estimated for each stratum, by the relative size of the strata to the population (see Equation 4).

The notations were defined as follow:

- N_k is the number of units in population belongs to stratum k
- n_k is the number of units in the sample belongs to stratum k
- w_i is the calculated weight
- $\bar{Y}_k$ is the mean of the target metric for stratum k

$$w_i = \frac{N_k}{n_k} \tag{4}$$

Finally, recalculate the estimates by following Equation 5:

$$\hat{Y}_{ps} = \sum_k \frac{N_k \bar{Y}_k}{N} \tag{5}$$

Regarding variance reduction, post-stratification reduces the variance of surveys by making the sample more representative and creating balanced strata, which should have a minor variance within subgroups (Valliant, 1993). Although stratified sampling reduces variance more effectively than post-stratification and simple random sampling, post-stratification can achieve similar variance reduction performance as stratification when dealing with large sample sizes (Xie & Aurisset, 2016).

2.4 Control Variates

Control variates is a common method to reduce variance in Monte Carlo simulations, first introduced by Boyle (1997). The approach assumes that by finding a random variable C, a control variate that is correlated with Y with a known expectation r, we can define a new estimator $\hat{Y}_{cv}$ (see Equation 6). θ can be any real number, but its optimal value is defined later:

$$\hat{Y}_{cv} = Y - \theta(C - r) \tag{6}$$

The variance of Y and the variance of $\hat{Y}_{cv}$ is slightly different:

$$var(\hat{Y}_{cv}) = var(Y) - 2\theta cov(Y, C) + \theta^2 var(C) \tag{7}$$

Based on Equation 7, the θ should be define as Equation 8 to achieve the minimum variance of $\hat{Y}_{cv}$:

$$\theta = cov(Y, C)/var(C) \tag{8}$$

By choosing the optimal θ, the minimum $var(\hat{Y}_{cv})$ will be:

$$var(\hat{Y}_{cv}) = var(Y)(1 - \rho^2) \tag{9}$$

Here, ρ is the correlation of Y and C, denoted as $cor(Y, C)$, showing that the greater the correlation between Y and C, the more variance can be reduced (Rubinstein & Marcus, 1985). The following two methods that will be introduced, CUPED and CUPAC, are based on the concept of control variate.

2.4.1 CUPED

CUPED (Controlled Experiments Utilizing Pre-Experiment Data) was first introduced by Deng and colleagues at Microsoft in 2013. It is widely used in the industry today due to its effectiveness in reducing variance and ease of implementation. Companies such as Netflix have experimented with

the performance of CUPED and stratification (Xie & Aurisset, 2016), and Booking.com has demonstrated the benefits of using CUPED (Jackson, 2018).

In CUPED, the covariate is set as the variate, denoted as C in the previous section. A covariate is a variable that correlates with the dependent variable but is not an independent variable; it is also independent of the treatment effect (Xie & Aurisset, 2016). Including covariates in the statistical model makes it more unbiased by eliminating confounding effects, which occur when another variable's effect is measured instead of the exposure's effect on the outcome (Jager et al., 2008). For example, age might be a covariate in an e-commerce study on engagement format changes, as it relates to engagement but not the format.

Choosing a covariate is essential since the statistics theory shows that finding a variate with a higher correlation to the target metric (Y) can reduce more variance. Deng et al. (2013) focused on selecting the covariate from pre-experiment data because that information is certainly not affected by the treatment effect, ensuring the independency of the experiment. Within trials, they suggested that setting the target metric's pre-experiment data as a covariate has a better ability to reduce variance. For instance, if testing a new feature's effect on revenue, the covariate would be the revenue before the A/B test.

CUPED effectively reduces variance, especially when minimal pre-experiment data is missing. Netflix's research showed a nearly 40% variance reduction in a simulated controlled experiment with existing users (Xie & Aurisset, 2016). Reduced variance brings benefits like more significant results and a reduced sample size requirement. Deng et al. (2013) found that using CUPED, an experiment's adjusted p-value, initially slightly below 0.05 after two weeks, was about 0.025 on day one. With CUPED, p-values are consistently smaller than the significance level, contrasting with the original results where p-values barely passed the significance threshold. Additionally, experiments can achieve similar p-value results with a smaller sample size. Deng et al. obtained a p-value of around 0.035 using

CUPED with half the users, compared to a p-value of approximately 0.005 in the original test with double the sample size.

CUPED performs worse when pre-experiment data is lacking, losing covariate information. For example, CUPED reduced only 1% of variance in a test with new users, compared to nearly 40% for existing users. Although post-stratification also performs worse with new users, CUPED is more significantly affected by this issue (Xie & Aurisset, 2016).

2.4.2 CUPAC

CUPAC (Control Using Predictions as Covariates), introduced by DoorDash, addresses CUPED's limitations when missing pre-experiment data (Tang et al., 2020). Instead of using pre-experiment data, CUPAC builds a machine-learning model to predict the covariate correlating with the target variable (Y). This approach includes various observation-level features independent of the treatment effect, providing more control variates than CUPED. In the machine learning model, selected variables predict the outcome value (Y), with predictions treated as the covariate in the CUPED model (Yang, 2023). This model maximizes the correlation between predictions and the target metric (Y), reducing the variance (Tang et al., 2020).

In DoorDash's research, CUPAC significantly enhanced the model's power, increasing from 0.095 to 0.138, higher than CUPED's 0.121 (Tang et al., 2020). This improvement is due to the high correlation between the variate and the target value, with CUPAC predictions correlating nearly 40% higher than pre-experiment data. Additionally, CUPAC shortens the time needed for an A/B test to achieve 80% power by almost 40% compared to the original model and 20% compared to CUPED. CUPAC performs better than CUPED when the most impactful covariate is not linearly related to the target value (Y). In a linear scenario, CUPED's mean squared error is six times greater than CUPAC's, and in a non-linear scenario, this difference increases to 116 times (Bonet, 2023).

Methodology

This chapter introduces the methodology for evaluating different variance reduction methods through simulated A/A tests. We conduct looped A/A tests on the same dataset using various tools: the baseline without advanced techniques, post-stratification, CUPED, and CUPAC models. After measuring their performance, we selected the most promising method to apply to real A/B testing data in the company.

2.2 Justification for Technique Selection

Based on the literature review, we summarized the advantages and disadvantages of each variance reduction technique in Table 2. Due to the high cost of conducting stratification, we developed models based on post-stratification, CUPED, and CUPAC. Building a stratification sampling system would require costly adjustments to the game server's sampling system and could disrupt currently running A/B tests in the company.

Table 2

Comparison of All Mentioned Variance Reduction Techniques

Technique	Description	Benefits	Limitations
Stratification	Divide the population into distinct strata and sample identical numbers of units from each stratum.	Reduces within-stratum variance.	Costly and time-consuming to implement.
Post-stratification	Similar to stratification but adjusts by adding weights after sampling.	Easier to implement than stratification.	Less effective than stratification.
CUPED	Controlling pre-experiment data as a covariate.	Effective and easy to implement.	Requires pre-experiment data, performs poorly on new user tests.
CUPAC	Estimate and control covariates with a machine learning model.	Handles complex situations better than CUPED.	More complex to implement than CUPED.

2.3 Simulated A/A Test

An A/A test is similar to an A/B test, but the treatment and control groups are identical, ensuring no difference between them. A/A tests are usually run before A/B tests to verify the reliability of the testing environment. If the A/B test system works correctly, there should be only a 5% chance of obtaining a p-value less than 0.05 in repeated rounds, indicating the system correctly identifies no difference between the groups (Kohavi et al., 2020).

Running an A/A test effectively evaluates variance reduction tools by measuring variances within the data. With two identical variants and no actual treatment effect, observed differences are due to random variations. Calculating the variance of these differences helps measure the natural variance of the data. Comparing models to the baseline, a smaller variance of differences indicates better performance of the variance reduction tool.

Researchers use A/A tests in the industry to evaluate variance reduction tools. Companies like Microsoft, Netflix, and DoorDash have implemented simulated A/A experiments for this purpose (Deng et al., 2013; Xie & Aurisset, 2016; Tang et al., 2020). These companies design their experiments differently; Microsoft ran a 3-week A/A test, while Netflix conducted repeated simulated experiments.

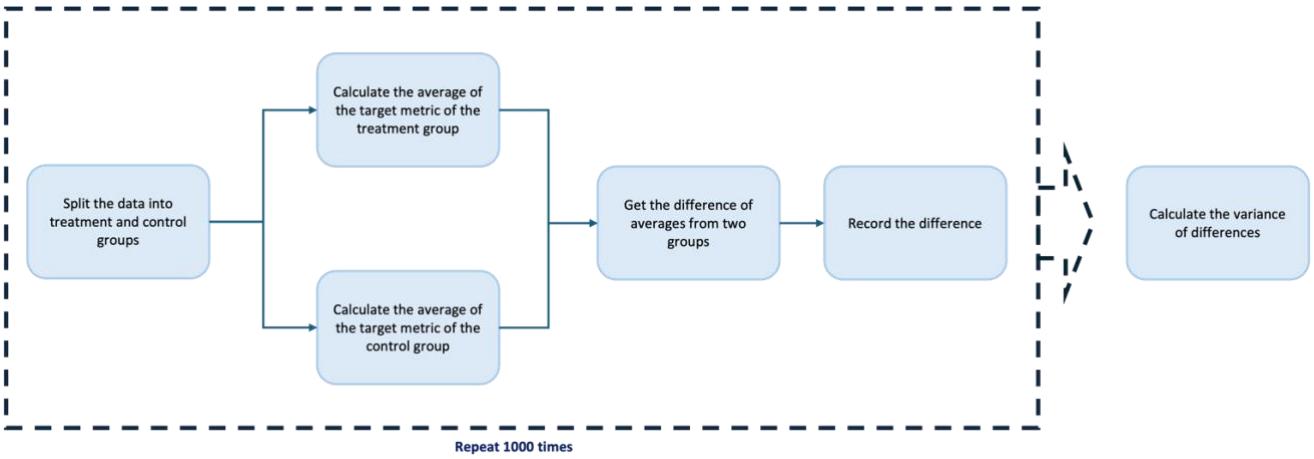
2.4 Experimental Design

Instead of running an A/A test over a period as Microsoft did, we decided to run a looped simulated test to save time. First, the data was run in an A/A test without any variance reduction techniques to establish the baseline variance. Then, multiple A/A tests were run using each variance reduction technique individually to obtain adjusted variances.

The structure of the A/A test is shown in Figure 2. Each A/A test was repeated 1,000 times. In each round, the dataset was randomly assigned into treatment and control groups, each comprising 50% of the data. We calculated the average of the target metric for both groups and compute the differences between these averages. The 1,000 differences should follow a normal distribution. Finally, we calculated the variance of these 1,000 differences and compared the variances obtained using different techniques.

Figure 2

Experiment Workflow for Simulated A/A Test with Variance Reduction Techniques



2.5 Data Source and Collection

All data used in this research comes from user activity data of Tripledot's popular card game, Solitaire. Tripledot collects and stores user activity and information on the game server, and we query the required data from the private database.

Using data from Solitaire offers several advantages. Firstly, its large user base provides a sufficient sample size, enhancing the experiment's reliability. Additionally, the comprehensive and diverse data allows for a more general test of variance reduction techniques, increasing the likelihood that these techniques can be applied to other company applications, enhancing the research's overall value.

All data is queried from Tripledot's Snowflake database using SQL. Thanks to effective ETL processes, the data from Snowflake is mostly structured and cleaned, requiring minimal preprocessing.

Data and Metric Selection

We queried four datasets for this project. Two datasets were used for simulated A/A tests to identify the most effective variance reduction technique. The other two datasets were used in the real-time A/B test section, where the selected techniques were implemented in an A/B test conducted by the product team to assess their impact.

4.1 Datasets for Simulated Tests

In the simulated A/A test section, historical user data was utilized instead of conducting real-time A/A tests that launched a new experiment on the server and assigned users to treatment and control groups in real-time. While this method may differ from other research, the historical data proved to be a reliable and beneficial resource for A/A testing. The core concept of an A/A test is randomization, which minimizes bias and ensures statistical equivalence between variants. This principle of random assignment was still achievable using historical data, providing a sense of reassurance and confidence in the A/A test. Furthermore, using historical data allowed for a larger sample size, as assigning all users to an A/A test in real time was impractical due to simultaneous A/B tests.

4.1.1 Features Introduction – Datasets for Simulated Tests

As mentioned, two datasets were used in this section. The second dataset extended the first one with more features and was applied to the most complex model, Model 7 (CUPAC with feature selection). Both datasets included all Solitaire players' vital information and activity from 1st April 2024, to 7th April 2024. We queried helpful factors from the database, and the columns of the two datasets are shown in Table 3 and Table 4. Dataset 2 had three additional columns compared to Dataset 1: the previous number of game wins, cost per install, and source. Both datasets had 5,635,310 rows, recording information of identical users who logged into the game from 1st April 2024 to 7th April 2024.

Table 3

Columns and descriptions of Dataset 1, which is used for Model 1 to Model 6

Feature	Description	Example
INSTALL_PK	The primary key in the table to refer to each user.	278
NUM_GAME_START	The total number of game start of each user from 1st April 2024 to 7th April 2024.	18
PLATFORM	The platform of the device.	ios
COUNTRY_CODE	The location where the user installs the game.	US
DEVICE_TYPE	What type of device the user is install the game on.	phone
PRE_NUM_GAME_START	The total number of game start of each user from 18th March 2024 to 31st March 2024.	50
COHORT_DAY	How many days the user has installed until 1st April 2024.	1185
INSTALL_DATE	The install date of the user.	2021-01-04

Table 4

Columns and descriptions of Dataset 2, which is used for Model 7

Feature	Description	Example
INSTALL_PK	The primary key in the table to refer to each user.	278
NUM_GAME_START	The total number of game start of each user from 1st April 2024 to 7th April 2024.	18
PLATFORM	The platform of the device.	ios
COUNTRY_CODE	The location where the user installs the game.	US
DEVICE_TYPE	What type of device the user is install the game on.	phone
PRE_NUM_GAME_START	The total number of game start of each user from 18th March 2024 to 31st March 2024.	50
COHORT_DAY	How many days the user has installed until 1st April 2024.	1185
INSTALL_DATE	The install date of the user.	2021-01-04
PRE_NUM_GAME_WON	The number of games won by the user from 18th March 2024 to 31st March 2024.	45
CPI	Cost per install.	0.5
SOURCE	Where and how the user learned about Solitaire and was directed to the install page.	Organic

4.1.2 Data Pre-processing - Datasets for Simulated Tests

In this section, we demonstrated the pre-processing flow with Dataset 2, since Dataset 1 is the sub-part of Dataset 2.

The number of missing values of Dataset 2 is shown as Table 5, but me barely handle them in this stage due to following reasons. First, missing values of previous number of game start and previous number of game won are handled differently in models we developed. Missing values of pre-experiment data play different roles in each variance reduction techniques, and we also tried different ways to handle missing values in each model to test how they affect the performance. Second, within the variables with missing values, we only use categorical variables like country and platform when building machine learning models. Missing values of categorical variables can still be processed by advanced machine learning like XGBoost. Therefore, we decided to keep that non-value and as much user data as possible.

Table 5

Number of Missing Values of Each Feature in Dataset 2

Feature	Number of Missing Values
INSTALL_PK	0
NUM_GAME_START	0
PLATFORM	1
COUNTRY_CODE	264
DEVICE_TYPE	0
PRE_NUM_GAME_START	1985338
COHORT_DAY	0
INSTALL_DATE	0
PRE_NUM_GAME_WON	1985338
CPI	0
SOURCE	0

Using one-hot encoding, we encoded the categorical columns (platform, country code, and device type). One-hot encoding transforms each categorical value into a new column that records binary numbers (0 and 1) to represent the original categorical values.

After removing rows with inconsistent install dates, which indicated unreliable information, we proactively identified new users who installed the game after the first day of our data period, 1st April 2024. We found 136 rows with install dates after 7th April 2024, which should not have been in our dataset and only includes users who played between 1st April 2024, and 7th April 2024. Our investigation revealed that this issue was likely due to the ETL process of the table storing installation data, which was recording the installation date incorrectly. It's important to note that the table recording user activity, including the number of game starts, was not affected. Therefore, we only removed those rows when using the install date information.

Labeling new and existing users is critical for analyzing player activity since new and old players usually behave differently. We labeled users who downloaded the game after April 1, 2024, as 1 (new user). We found that identifying new users based solely on missing pre-experiment data was not reliable. If an existing user did not play the game during the pre-experiment period, their pre-experiment data would still be missing, even though they are not new players.

Finally, we calculated the previous winning rate by dividing the previous number of games won by the previous number of games started. This transformation helps to normalize the data, ensuring that the variable's scale is consistent with others. Additionally, we conducted feature selection while training machine learning models, and the winning rate might be more informative and provide a more accurate measure of skill or success.

To sum up, both Dataset 1 and Dataset 2 include an extra column to indicate whether a user is new or existing, and all categorical variables are transformed into multiple columns with Boolean values. Dataset 2 includes an additional column to record the winning rate.

4.2 Data for Practical Implementation

In the real-time A/B test section, we will use data from recent A/B tests conducted by the product team. Since we applied variance reduction techniques to two different A/B tests, we queried two datasets. Both datasets have identical columns, as shown in Table 6. More details about the content and the A/B tests are explained in the section on practical implementation. The pre-processing process is the same as that used for the data in the simulated tests, which are Dataset 1 and Dataset 2.

Table 6

Columns and descriptions of Datasets for Practical Implementation

Feature	Description	Example
INSTALL_PK	The primary key in the table to refer to each user.	278
VARIANT_NAME	The variant to which the user is allocated.	control
VARIANT_DEFAULT	0 for all treatment groups; 1 for the control group	1
ASSIGNED_DATE	The date that the user is assigned to the A/B test.	2024-06-29
NUM_GAME_START	The total number of games the user starts during the A/B test.	18
TOTAL_REVENUE	Total revenue that the player makes during the A/B test.	0.638
NUM_AD_IMPRESSION_REWARDED	The number of ad impression show for rewards during the A/B test	14
NUM_AD_IMPRESSION_INTERSTITIAL	The number of impressions of interstitial ads during the A/B test	23
NUM_AD_IMPRESSION_BANNER	The number of impressions of banner ads during the A/B test.	90
PRE_NUM_GAME_START	The total number of games the user started before 0 to 14 days of assignment to the A/B test.	50
PRE_TOTAL_REVENUE	Total revenue that the user generated before 0 to 14 days of assignment to the A/B test.	0.322
PRE_NUM_AD_IMPRESSION_REWARDED	The number of ads the player saw for rewards before 0 to 14 days of assignment to the A/B test.	3

PRE_NUM_AD_IMPRESSION_INTERSTITIAL	The number of interstitial ads the player saw before 0 to 14 days of assignment to the A/B test.	19
PRE_NUM_AD_IMPRESSION_BANNER	The number of banner ads the player saw before 0 to 14 days of assignment to the A/B test.	176
PLATFORM	The platform of the device.	ios
COUNTRY_CODE	The location where the user installs the game.	US
DEVICE_TYPE	What type of device the user is install the game on.	phone
COHORT_DAY_INSTALL	How many days the user has installed until the assignment data to the A/B test.	1593
INSTALL_DATE	The install date of the user.	2020-02-18
PRE_NUM_GAME_WON	The number of games won by the user before 0 to 14 days of assignment to the A/B test.	45
CPI	Cost per install.	0.5
SOURCE	Where and how the user learned about Solitaire and was directed to the install page.	Organic

4.3 Target Metric Selection

One specific metric must be chosen for all A/A tests, similar to the target variable in an A/B test. In this research, we chose the number of new game starts as the primary metric, a standard metric in A/B tests by the product team. The number of new game starts indicates user engagement, which is crucial for developing a mobile game. Higher game starts suggest greater player interest, leading to game improvements. In the A/A test section, we calculated the average number of game starts for the treatment and control groups to get the average treatment effect, which should be zero in the A/A tests.

To evaluate the performance of different variance reduction techniques, we will calculate the percentage of variance reduction in the differences between variants in the A/A test. In the real-time A/B test section, we will also examine the adjusted p-value and confidence interval.

Model Design and Simulation

In this chapter, we demonstrate the implementation of different variance reduction techniques and present their comparative results. Seven models were implemented: the baseline, post-stratification, two versions of CUPED, and three versions of CUPAC. We designed and developed more advanced models for CUPED and CUPAC, customizing them from the basic versions due to their high potential.

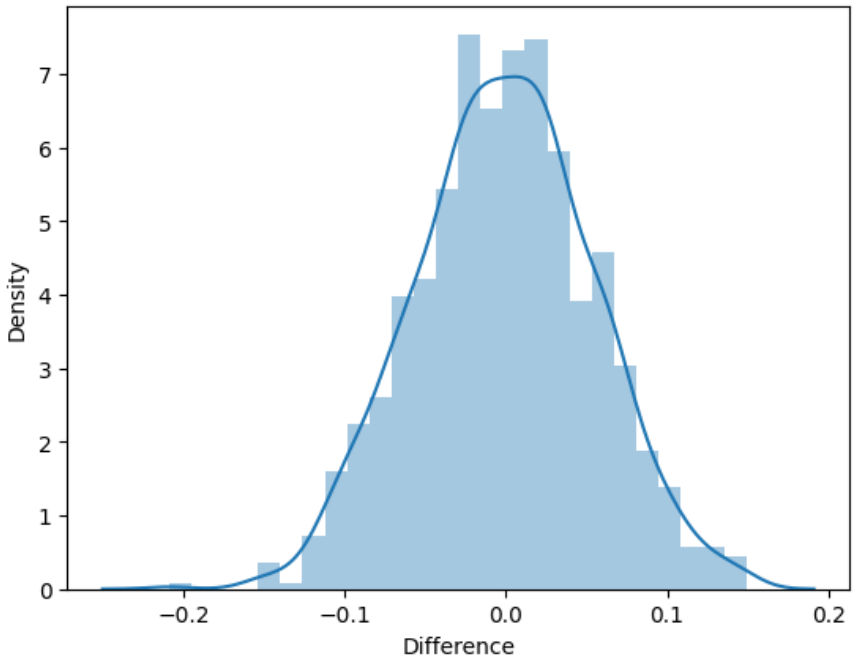
5.1 Model 1 - Baseline

The baseline model implemented an A/A test without any variance reduction technique, showing all users' natural variance and distribution of game starts. This model was compared to others to evaluate variance reduction in following sections.

After 1,000 rounds of the A/A test, the results were shown in Figure 3, illustrating the difference distribution between treatment and control groups in each round. The distribution was normally distributed and centered around 0, indicating no difference between variants. The average game start differences mostly ranged from -0.2 to +0.2, with a variance of 0.002997. We aimed to see if a variance reduction technique can narrow this range.

Figure 3

Distribution of Differences Between Variants in the Baseline A/A Test



Note. The x-axis represented the difference of average of number of game start between treatment and control groups. The y-axis represented the density that indicates the frequency of difference values occur in the dataset.

5.2 Model 2 - Post-stratification

The concept of post-stratification ensured that the sample reflected the population by weighting estimators based on the population size ratio to sample size within each stratum. Therefore, the first step was to allocate users to strata.

The stratification variable should relate to the target variable. Hence, we categorized users based on their engagement, which was directly related to the number of game starts. We labeled users from 1 to 4 (easy, casual, moderate, and hardcore players) based on the 25th, 50th, and 75th percentiles of game starts (1, 4, and 20, respectively). For example, if a user played ten new games, they were

allocated to category 3 (moderate players). After the allocation process, all users were assigned to different strata.

Based on Equation 5, we defined two functions to calculate the mean difference between variants and the weighted mean. First, we calculated the weights for each stratum, defined as the ratio of the population size of the stratum to the sample size of the stratum. For example, if there were 10,000 people in the population classified as easy players (stratum 1) and 2,000 users in the treatment group who are easy players, the weight for this stratum would be calculated as $\frac{10{,}000}{2{,}000} = 5$.

Next, we calculated the adjusted mean for each treatment and control group stratum by applying the weights to the original means. The adjusted means for each stratum were aggregated, and the total was divided by the sum of all strata's weights to obtain the stratified mean difference between the treatment and control groups.

Figure 4 shows the result of running 1,000 rounds of A/A tests with post-stratification. In each round, the dataset was split into treatment and control groups, and the weighted mean difference was calculated. The results were normally distributed with a peak around 0, indicating no significant difference. The differences mostly ranged from -0.15 to +0.15, narrower than the baseline range of -2.0 to +2.0. Table 7 shows that the variance of this distribution was 0.001997, smaller than the baseline variance of 0.002997, indicating that the post-stratification model reduced variance by 33.37%.

Figure 4

Distribution of Differences Between Variants in the A/A Test with post-stratification

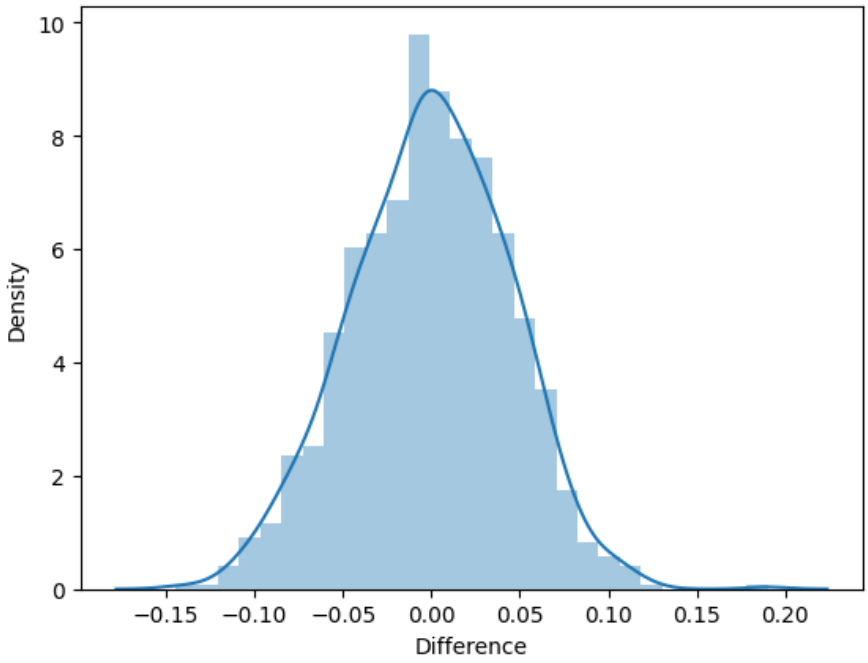


Table 7

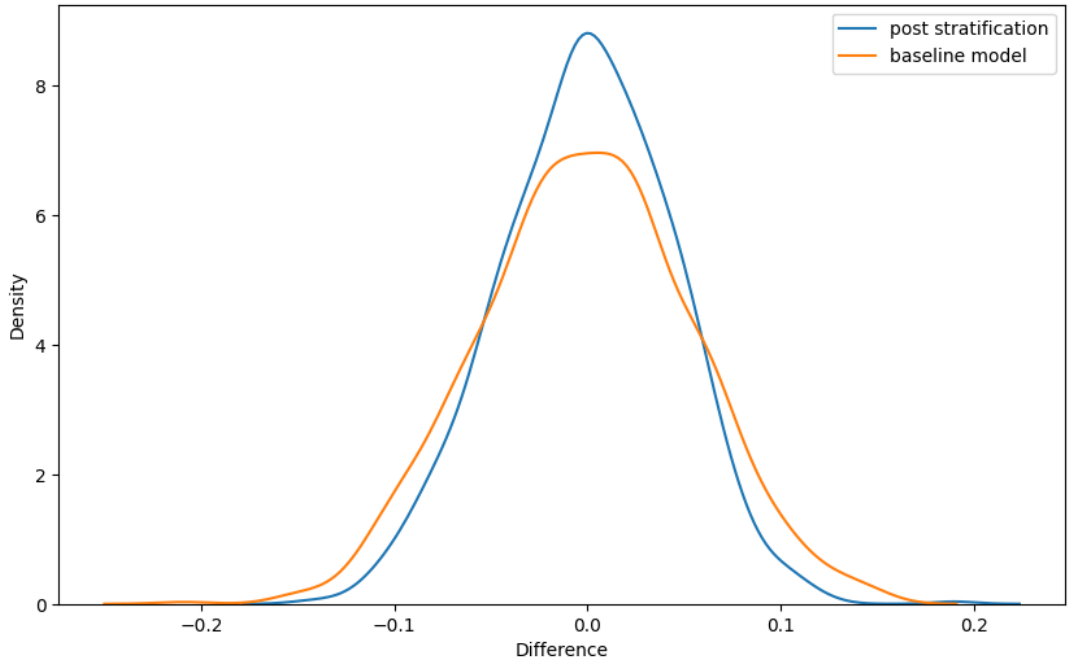
Comparison of Critical Metrics of Baseline and Post-stratification models

	Variance	**Variance Reduction Percentage Comparing to the Baseline Model**
Baseline model	0.002997	0.0%
Post-stratification	0.001997	33.3726%

Post-stratification helped reduce variance, enhancing the A/B test's ability to capture more minor treatment effects. In Figure 5, the curve lines show the distributions of the A/A test results for the baseline and post-stratification models. The line representing the post-stratification model was narrower and had a higher peak than the baseline, indicating a smaller variance.

Figure 5

A/A Test Results comparison - Baseline and Post-stratification Models



Note. This figure is a combination of Figure 3 and Figure 4, displaying the distribution curves of two A/A test results.

5.3 Model 3 & 4 - CUPED

CUPED is a covariate controlling method used to reduce variance in A/B tests by adjusting the estimator with pre-experiment data of the target variable. Two critical settings in CUPED are handling missing pre-experiment data and choosing the pre-experiment period. We developed two models for handling missing data differently: Model 3 left rows with missing data unadjusted, while Model 4 used a machine learning model to predict missing values.

For the pre-experiment period, we used data from 14 days before the A/A test, as Deng et al. (2013) recommended. The 14-day duration balanced the risk of having too many missing values and mismatching. We used user data from 18th March 2024 to 31st March 2024, with the A/A test starting on 1st April 2024.

The adjusted CUPED value was calculated in both models, and 1000 rounds of A/A tests were run to check variance reduction. We calculated the optimal θ, which will reduce most variance of the target variable, and then calculated the CUPED adjusted target variable for the A/A test. The pseudo code for CUPED adjustment is as follow:

Algorithm 1 CUPED Adjustment

Input: original data

Output: data with an extra column with CUPED adjusted number of game start

```
Function Cuped (data):
    covariance = covariance of num_game_start and pre_num_game_start
    θ = covariance / variance of pre_num_game_start
    cuped_num_game_start = num_game_start - θ*(pre_num_game_start - mean of
    pre_num_game_start)
return data
```

5.3.1 Model 3 – Basic CUPED

Among the 5,635,310 users, 1,985,338 did not have pre-experiment data, resulting in approximately 35% missing values for the previous number of game starts. In Model 3, we handled these missing values by leaving the number of game starts for these players unadjusted, a method similar to Jackson's approach at Booking.com (2018). Jackson's research found that filling missing pre-experiment data with the sample mean produced results equivalent to not adjusting the target value with CUPED, a key step in our data analysis process.

For example, if a user downloaded the game on the first day of the A/A test, April 1, 2024, the user would not have any pre-experiment game activity records, and the adjusted number of game starts would be the same as the actual starts during the A/A test. Conversely, if a player had pre-experiment records, the adjusted number of game starts would differ due to the CUPED calculation.

The dataset was processed with the CUPED function to get CUPED-adjusted game starts for every row. We then used the CUPED-adjusted game starts as the target variable in the A/A test. The A/A test

results, shown in Figure 6, exhibit a normal distribution centered at 0, with most differences between the treatment and control groups within -0.1 and +0.1.

Figure 6

Distribution of Differences Between Variants in the A/A Test with Basic CUPED Model

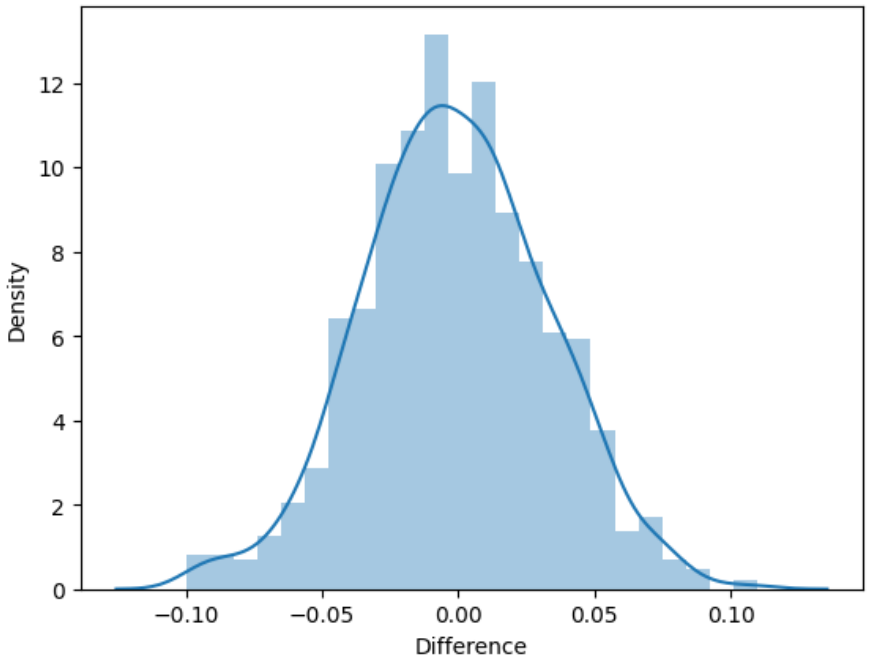


Table 8 and Figure 7 show that Model 3, the basic CUPED model, effectively reduced variance in an A/A test environment, improving the sensitivity of A/B tests. Table 8 compares the variance to the baseline model, indicating that Model 3 reduced variance by 61%. In Figure 7, the distribution of differences between the two variants with the basic CUPED model was narrower and had a higher peak.

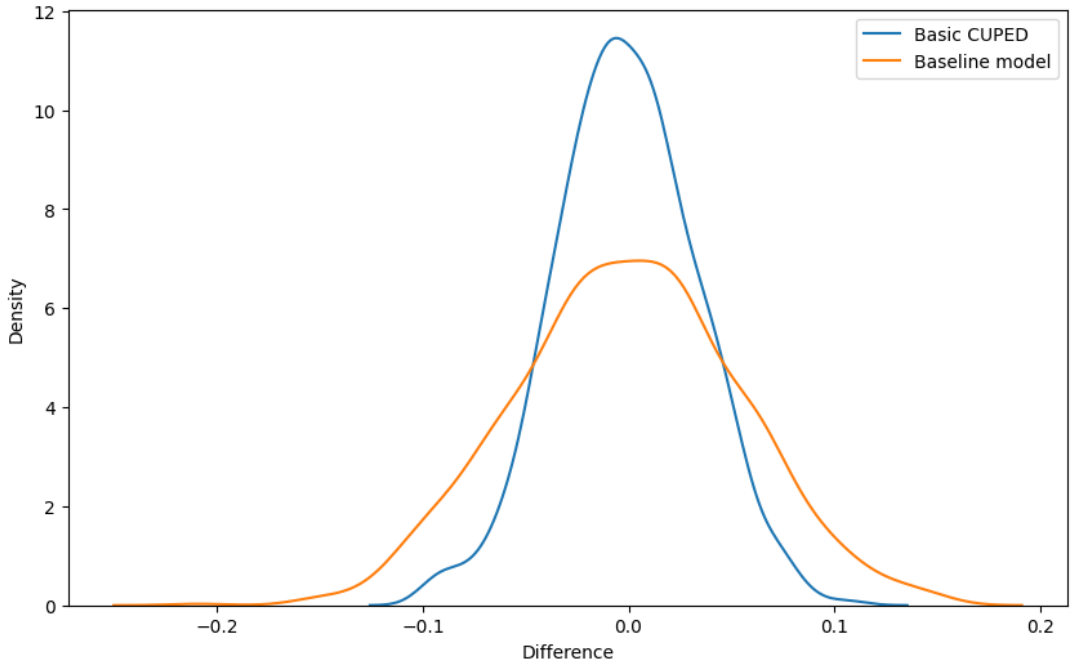
Table 8

Comparison of Critical Metrics of Baseline and the Basic CUPED models

	Variance	**Variance Reduction Percentage Comparing to the Baseline Model**
Baseline model	0.002997	0.0%
Basic CUPED	0.001154	61.480803%

Figure 7

A/A Test Results comparison - Baseline and Basic CUPED Model



Note. This figure is a combination of Figure 3 and Figure 6, displaying the distribution curves of two A/A test results.

5.3.2 Model 4 – CUPED with Machine Learning

After implementing Model 3, we realized that handling missing values more effectively could improve variance reduction by making the pre-experiment information more complete. Model 4 included a machine learning model that predicted new users' previous number of game starts. The CUPED adjustment then used all pre-experiment data, including actual data for existing users and predictions for new users.

The training dataset consisted of existing users, and the testing dataset included new users. The target variable was the number of games that started before the A/A test. Five independent variables were used: the number of game starts during the A/A test, platform, country code, device type, and cohort day. The machine learning model learned the relationship between these features and the

number of games that start before the A/A test of existing users to predict those data for new users so that we handle missing values of a previous number of games starting for new users.

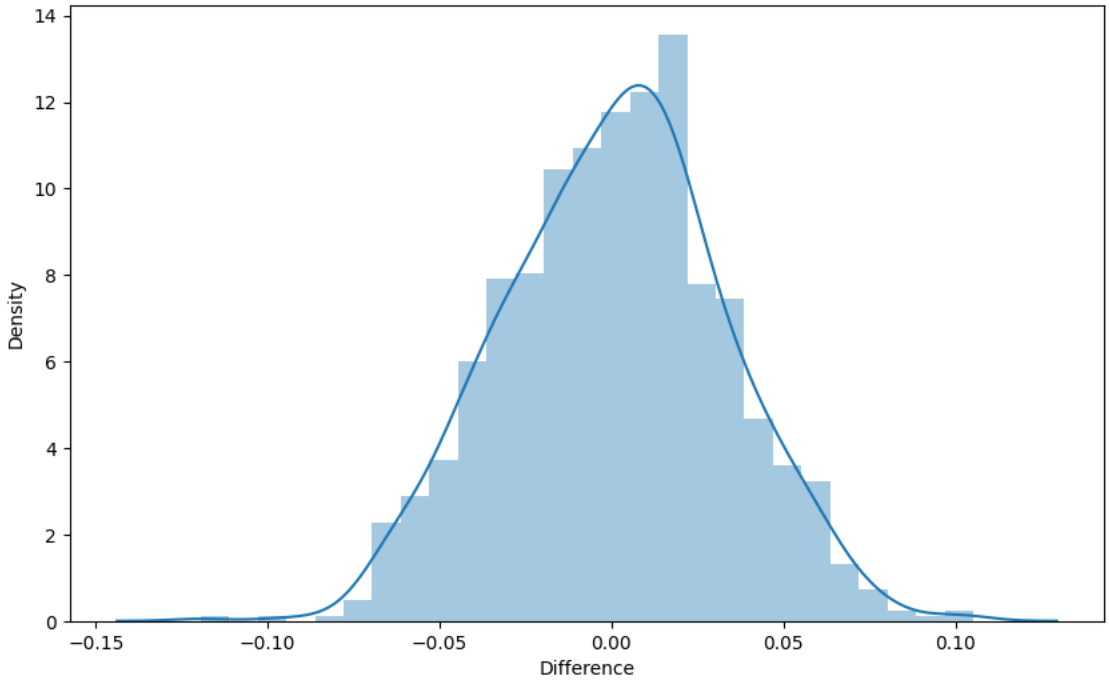
We used Dataset 1, which includes columns: install primary key, number of game starts, platform, country code, device type, number of game starts before the A/A test, cohort day, install date, and new user. First, we split the dataset into training and testing sets based on user status. Existing users were allocated to the training set, and new users were allocated to the testing set. Missing pre-experiment game starts for existing users were filled with 0, indicating no game activity between 18th March 2024 and 31st March 2024.

Next, we handled missing values for new users using Extreme Gradient Boosting (XGBoost), a powerful machine learning model suitable for complex, large-scale data. XGBoost was trained on the training set (existing users) and predicted the pre-experiment game starts for the testing set (new users).

After combining the training and testing sets, we had a complete dataset without missing pre-experiment game starts. We then applied the CUPED adjustment function to the entire dataset to obtain the CUPED-adjusted target variable. Finally, we ran the A/A test 100 times, and the results in Figure 8 showed that the mean differences between the two variants were normally distributed, peaking around 0. Most differences ranged between -0.1 and +0.1, similar to Model 3's range.

Figure 8

Distribution of Differences Between Variants in the A/A Test with CUPED with ML



However, as shown in Table 9 and Figure 9, Model 4 (CUPED with ML) reduced more variance than Model 3. Model 4 achieved a variance reduction of 65.4%, compared to 61.4% for Model 3. The distribution of results from Model 4 is narrower and higher but slightly skewed, indicating a slight positive bias that might affect the practical A/B test's accuracy in detecting treatment effects. Table 9 shows that the mean difference for Model 4 is 189.1% compared to the baseline, while Model 3 is -28.5%, suggesting that Model 4 has a greater impact on changing the mean. This bias could be due to overfitting or the unsuitability of the covariate, implying that predicting pre-experiment data for new users might not work as well as expected.

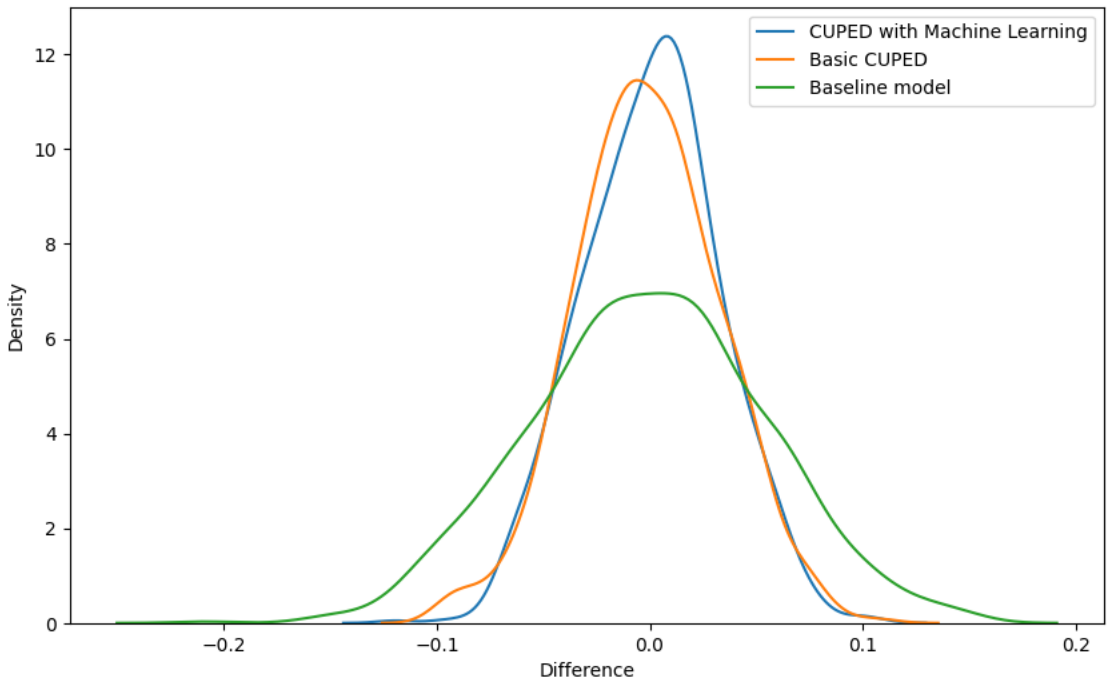
Table 9

Comparison of Critical Metrics of Baseline and the CUPED with ML models

	Variance	Variance Reduction Percentage Comparing to the Baseline Model	Mean	Mean Diff Comparing to the Baseline Model
Baseline model	0.002997	0.0%	-0.000953	0.0%
Basic CUPED	0.001154	61.480803%	-0.001225	-28.546257%
CUPED with ML	0.001037	65.405063%	0.000850	189.143956%

Figure 9

A/A Test Results Comparison – Baseline, Basic CUPED Model, and CUPED with ML



Note. This figure is a combination of Figure 3, Figure 6, and Figure 8, displaying the distribution curves of two A/A test results.

5.4 Model 5 to 7 – CUPAC

CUPAC, an advanced technique based on CUPED, integrates observational features related to the experiment into the covariate using a machine learning model, instead of a single pre-experiment

data point. The process of building a CUPAC model involved several vital steps. First, we selected features to train the machine learning model. Then, we trained the model to predict the estimated covariate. Next, we implemented the CUPED adjustment function on the predicted covariate and target variable. Finally, we ran 1,000 rounds of A/A tests on the CUPED-adjusted target variable.

The machine learning model played different roles in Model 4 (CUPED with ML) and the CUPAC method. In Model 4, the covariate was the number of games that started before the A/A test, and the machine learning model handled missing data for new users. In Models 5, 6, and 7, the machine learning model created the covariate from multiple selected features. We used XGBoost for all models, as Tang et al. (2020) found gradient-boosted trees effective in identifying suitable covariates.

Before training all CUPAC models, the dataset was not split into training and testing sets because the goal was to find the relationship between features and the target variable within each A/B test dataset. Overfitting was not a concern, as each CUPAC model was trained for specific A/B tests. For real-time A/B test analysis, a new model must be trained for each test to identify the most suitable covariate.

We developed three CUPAC models. Model 5 was the simplest, using the fewest features to aggregate the covariate. Model 6 included more features and Model 7 used the most features with feature selection during training.

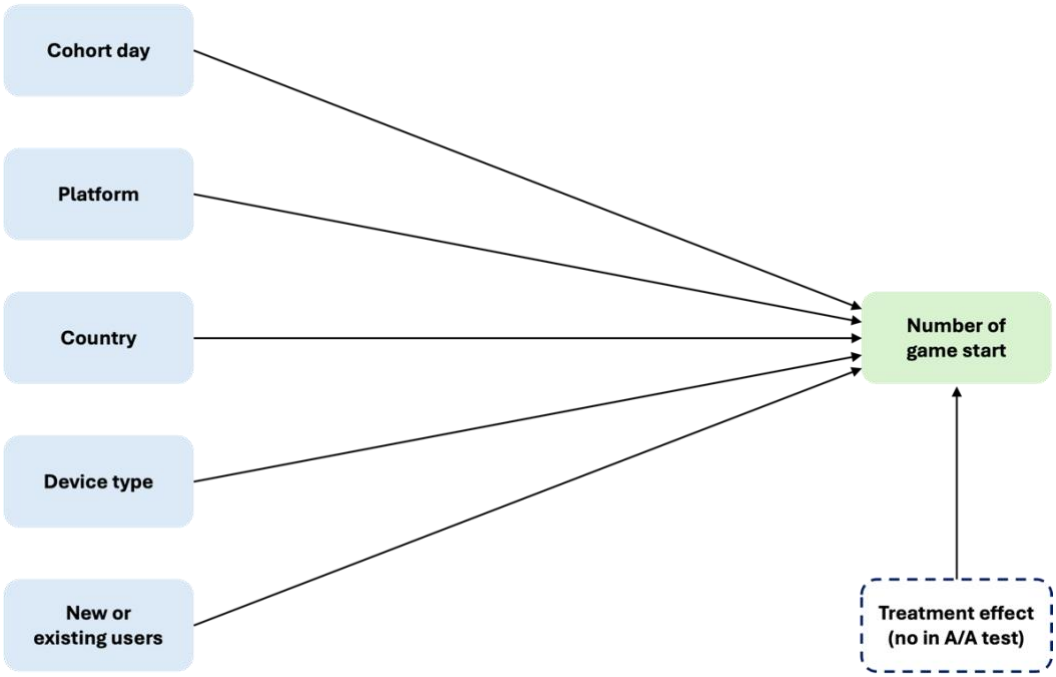
5.4.1 Model 5 – CUPAC without Pre-experiment Data

For the first trial of developing CUPAC, we selected five observational features describing each user's background to estimate the covariate. These features had to be independent of the treatment effect but could provide additional information to build a covariate. Figure 10 shows that the five selected factors are qualified. These factors could influence the player's behaviour, affecting the number of game starts, but were not influenced by the treatment effect in A/B tests. For instance, in an A/B test to determine whether the game layout affects the number of game starts, we can control

for factors like cohort day and platform, which also impact game starts. By controlling these factors with CUPAC, we could accurately measure the treatment effect due to the layout change.

Figure 10

The DAG of Selected Features in Model 5 and Number of Game Start



After the dataset was preprocessed with cleaning and encoding, we selected the five factors above as the independent variable and the number of games that started during the A/A test as the dependent variable. We trained an XGBoost model on the entire dataset by feeding and used the model to predict by feeding the five factors above to predict the target variable, the number of game starts during the A/A test. The predictions we got are the estimated covariates. Finally, we calculated the CUPED-adjusted number of game starts and ran the A/A test. Figure 11 shows the result of the A/A test; the normal distribution is centred around 0 and ranges roughly between -2.0 and +2.0.

Figure 11

Distribution of Differences Between Variants in the A/A Test with CUPAC without Pre-experiment Data

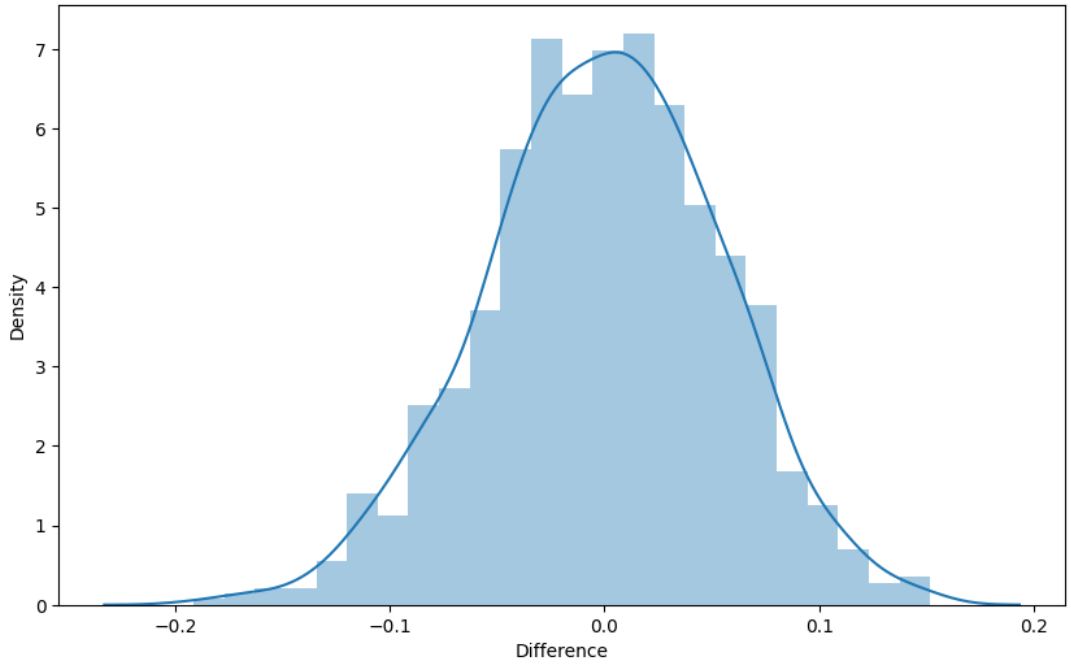


Table 10 and Figure 12 show that Model 5 did not reduce variance. The variance using Model 5 was 1.54%, higher than the baseline variance, and the two distributions nearly overlapped. This result suggested that the features were incorrectly selected, leading to an estimated covariate that was not correlated with the target value. Consequently, the covariates were not effectively controlled.

Table 10

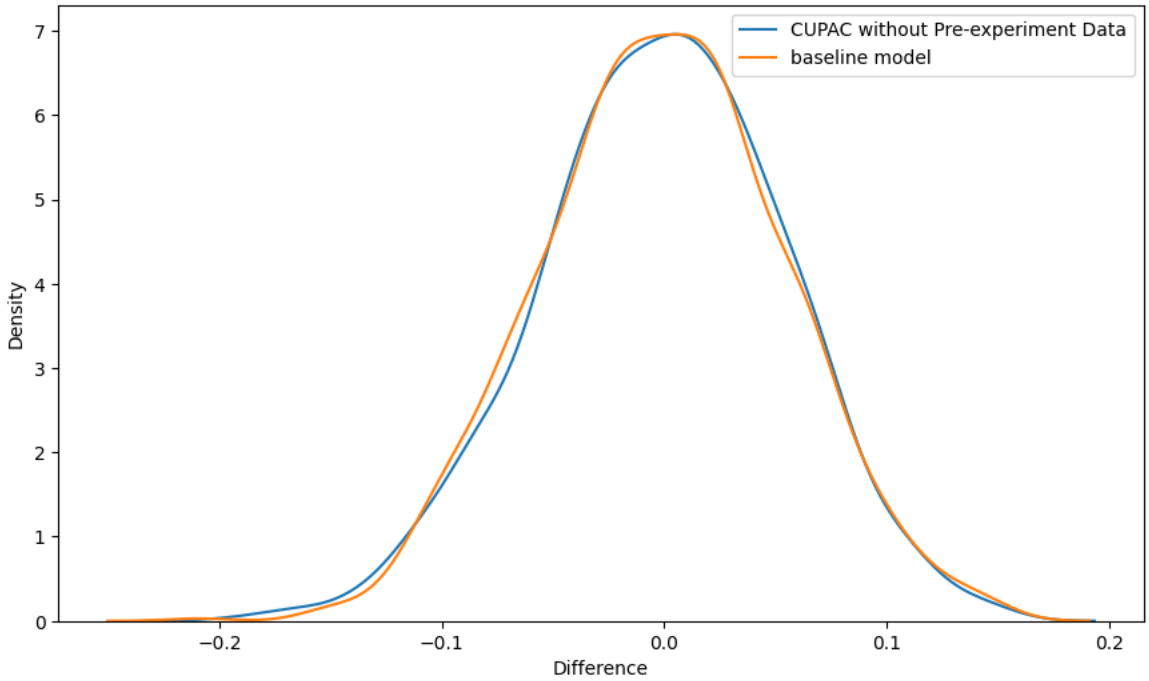
Comparison of Critical Metrics of Baseline Model and CUPAC without Pre-experiment Data

	Variance	Variance Reduction Percentage Comparing to the Baseline Model	Mean	Mean Diff Comparing to the Baseline Model
Baseline model	0.002997	0.0%	-0.000953	0.0%
CUPAC without Pre-experiment data	0.003043	-1.542138%	-0.000866	9.115101%

Note. The variance reduction percentage should be positive if variance was reduced, indicating the mean uplift compared to the baseline. A positive mean difference percentage shows that the mean of the A/A test result is higher than that of the baseline model.

Figure 12

A/A Test Results Comparison – Baseline and CUPAC without Pre-experiment Data



Note. This figure is a combination of Figure 3 and Figure 11, displaying the distribution curves of two A/A test results.

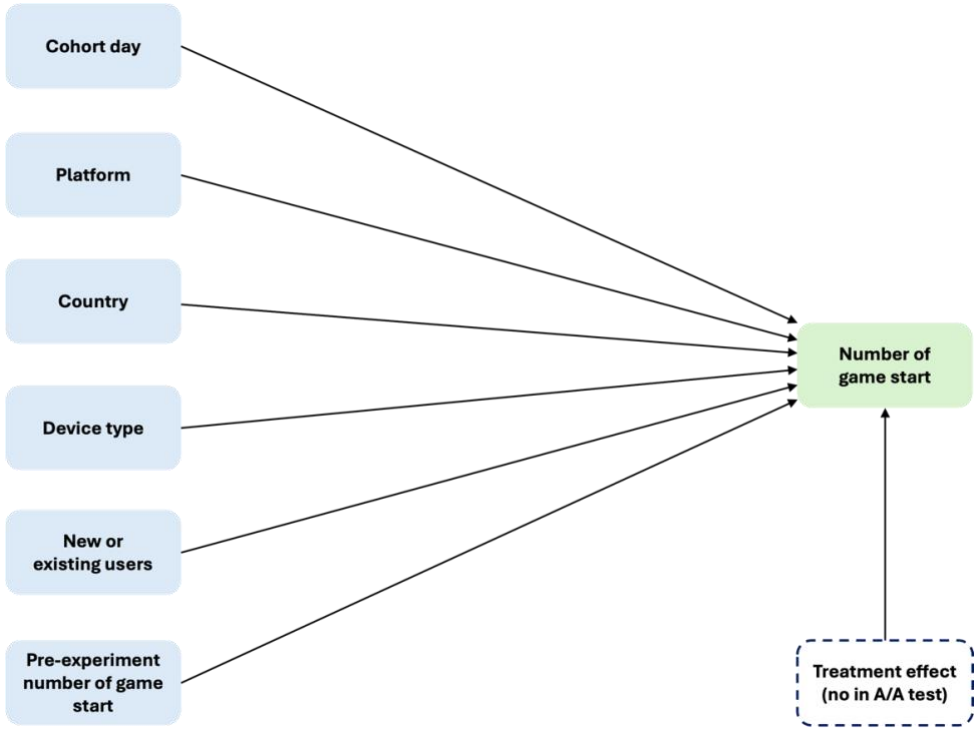
5.4.2 Model 6 – CUPAC with Pre-experiment Data

After the failure of developing Model 5, we sought to enhance the covariate quality. Learning from CUPED, we developed Model 6 by including pre-experiment information. Model 6 added the pre-experiment number of game starts to estimate the covariates, compared to Model 5. The Directed Acyclic Graph (DAG) for Model 6, shown in Figure 13, indicated that the six selected factors—cohort day, platform, country, device type, new or existing user, and pre-experiment information—might

affect the target value. Therefore, we controlled these six factors by merging them into an estimated covariate.

Figure 13

The DAG of Selected Features in Model 6 and Number of Game Start



To include the pre-experiment number of game starts in the model, we addressed the missing values for users who had not installed or played between March 18, 2024, and March 31, 2024, by filling all missing values with 0. This resulted in scenarios where the pre-experiment number of game starts is 0 for both new and inactive existing users. However, labelling new and existing users allows the machine learning model to identify the relationship between pre-experiment data and user status.

The XGBoost model was trained with the six selected factors to predict the number of game starts. We then calculated the adjusted CUPED value with the estimated covariate and the actual number of game starts, and ran 1,000 A/A tests with the CUPED-adjusted number of game starts. Figure 14 shows the results of running Model 6, with differences normally distributed and ranging

between -0.1 and +0.1, narrower than Model 5's range of -2.0 to +2.0. The peak around 0 indicated that the A/A test is not biased.

Figure 14

Distribution of Differences Between Variants in the A/A Test with CUPAC with Pre-experiment Data

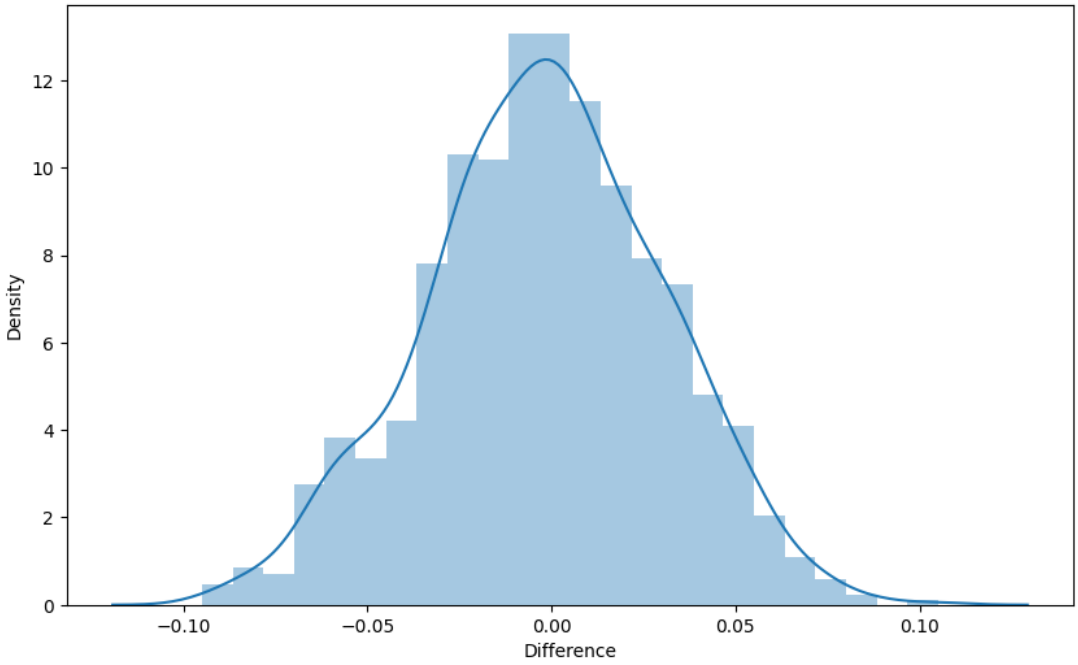


Table 11 and Figure 15 show that Model 6 had a solid ability to reduce variance. In Table 11, Model 6 reduced variance by 65.2% compared to the baseline, and the distribution of the A/A test was much narrower in Figure 15. However, Model 6 still impacted the accuracy of detecting the treatment effect. According to Table 11, the means of the baseline and Model 6 were slightly different.

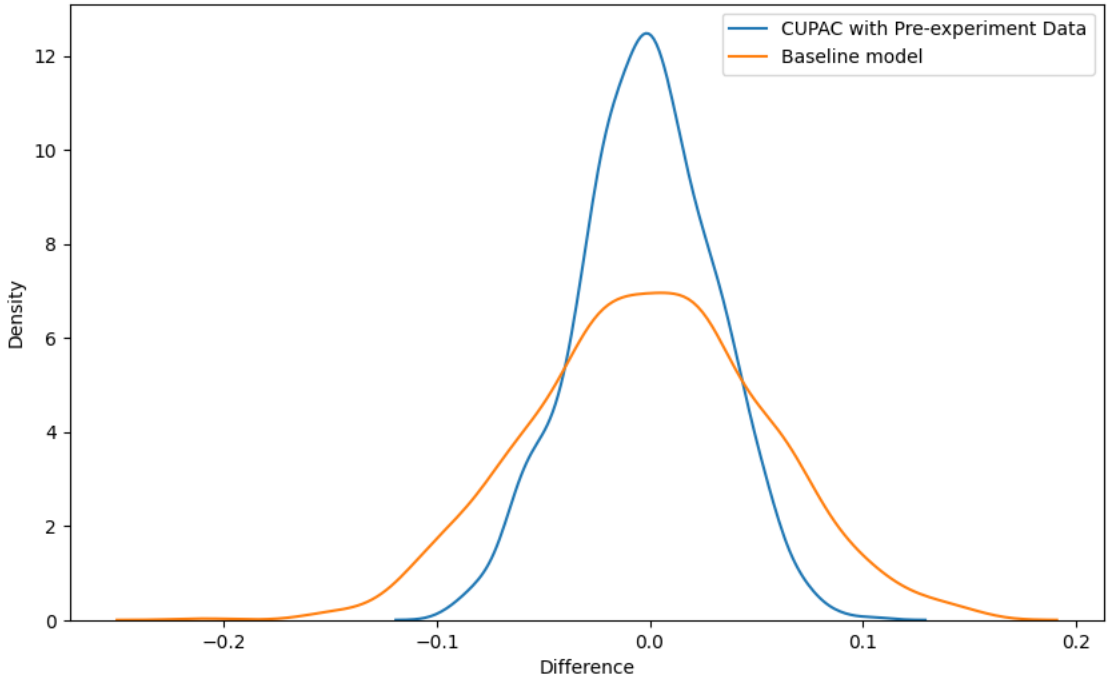
Table 11

Comparison of Critical Metrics of Baseline Model and CUPAC with Pre-experiment Data

	Variance	Variance Reduction Percentage Comparing to the Baseline Model	Mean	Mean Diff Comparing to the Baseline Model
Baseline model	0.002997	0.0%	-0.000953	0.0%
CUPAC with pre-experiment data	0.001041	65.271019%	-0.002028	-112.778525%

Figure 15

A/A Test Results Comparison – Baseline and CUPAC with Pre-experiment Data



5.4.3 Model 7 – CUPAC with Feature Selection

Model 7, an extension of the successful Model 6, incorporated a different dataset, Dataset 2, and introduced three new columns: source, CPI (cost per install), and pre-experiment number of

winning games. Figure 16 illustrates the relationship between these features and the number of game starts, demonstrating the significant influence of these features on user behaviour.

A higher CPI indicates that the app targets higher-quality users more likely to be interested in the game, increasing their engagement. The source factor also provides critical information about the user's background; users who find the app through organic search on app stores are often actively looking for a game and might play more. The pre-experiment number of winning games can also affect user behaviour during the A/A test, as a lower winning rate might indicate user frustration and potential churn.

As in Model 6, we filled all missing values for the pre-experiment number of game starts and the number of winning games with 0 in Model 7. Then, we transformed the number of winning games into a winning rate by dividing by the number of game starts, allowing for a standardized comparison across different players.

Model 7 involved training two XGBoost models. We trained the first machine learning model to obtain the feature importance scores, which evaluated the importance of each feature in the model. Figure 17 shows the top 20 essential features, highlighting that the pre-experiment number of game starts plays a critical role in model training. We then calculated the median importance score and set it as the threshold, filtering out the less important features to reduce model noise. This process filtered out 151 features, leaving 151 features for the final model.

Figure 16

The DAG of Selected Features in Model 7 and Number of Game Start

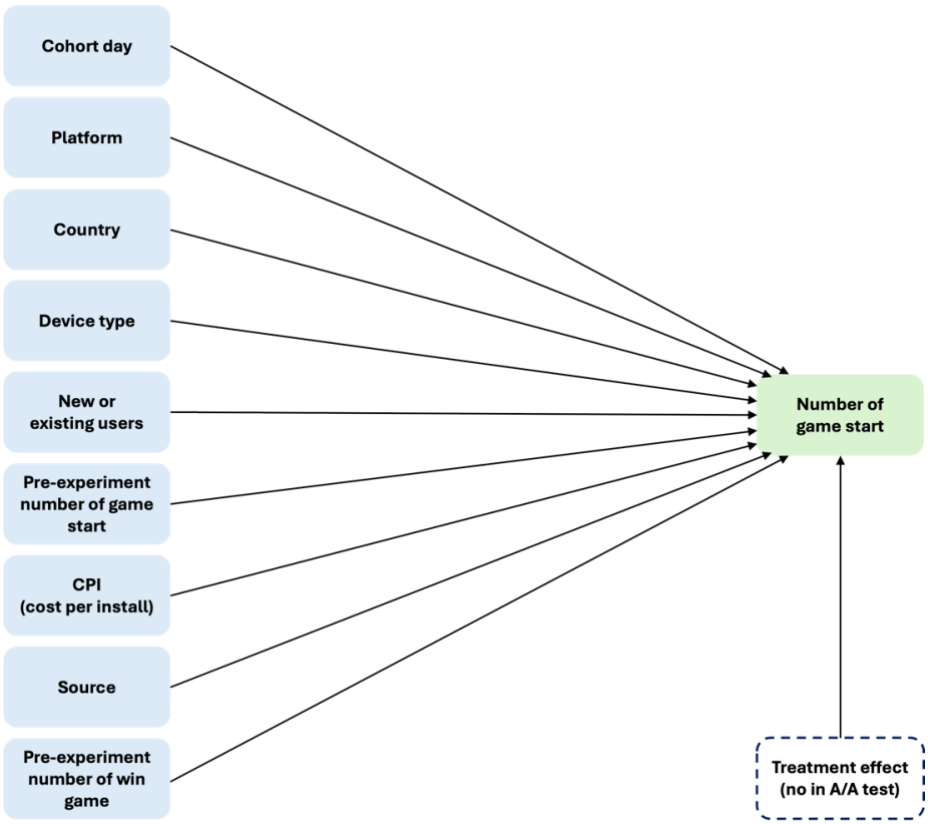
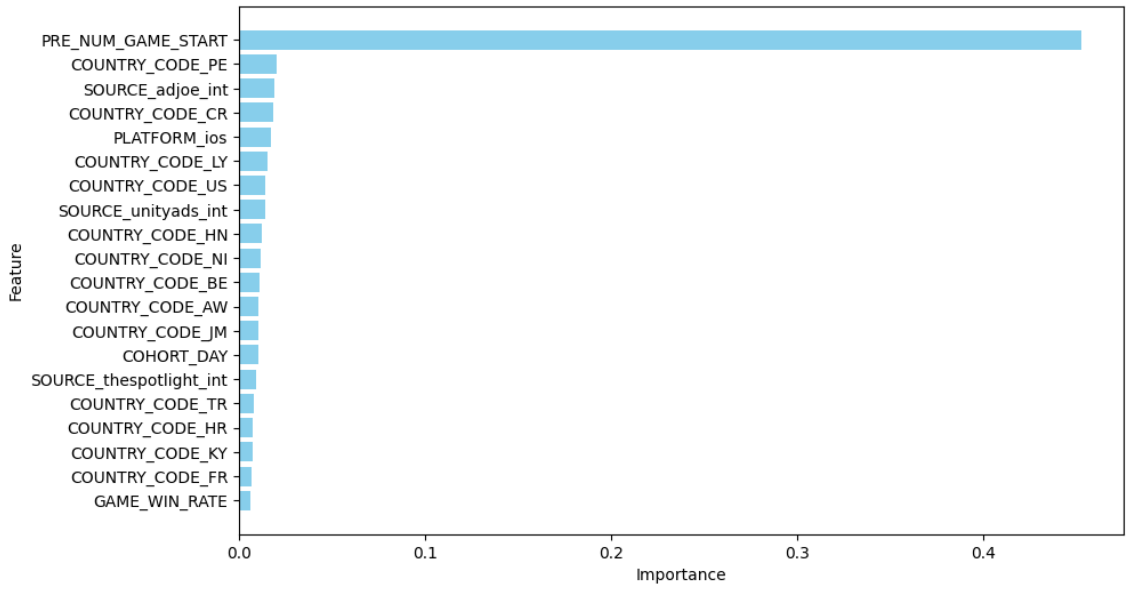


Figure 17

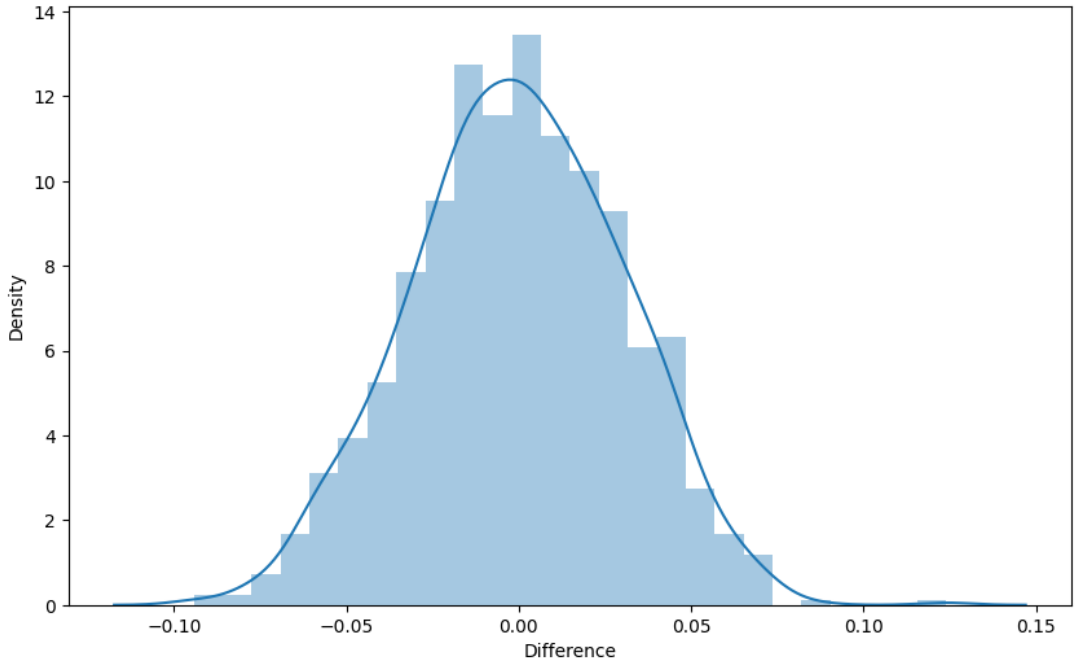
Top 20 Important Feature in the Machine Learning Model of Model 7



After selecting essential features, we retrained the XGBoost model using these features as independent variables, and the number of the game starts as the dependent variable to obtain the estimated covariate. We then calculated the adjusted CUPED number of game starts and ran the A/A test for 1,000 rounds. The result, shown in Figure 18, displayed a normal distribution with minimal bias and a range between -0.1 and +0.1.

Figure 18

Distribution of Differences Between Variants in the A/A Test with CUPAC with Feature Selection



Overall, the results of Model 7 perform slightly better than Model 6. From Table 12, we can conclude that Model 7 reduces more variance than Model 6, with a reduction of 68.6% compared to 65.2%. The mean change is also smaller for Model 7 than for Model 6, which is preferred. However, in Figure 19, the distributions of Model 6 and Model 7 almost overlap, and Model 6 even appears to have a higher peak. This implies that Model 6 might lead to a more concentrated distribution, even with a larger variance.

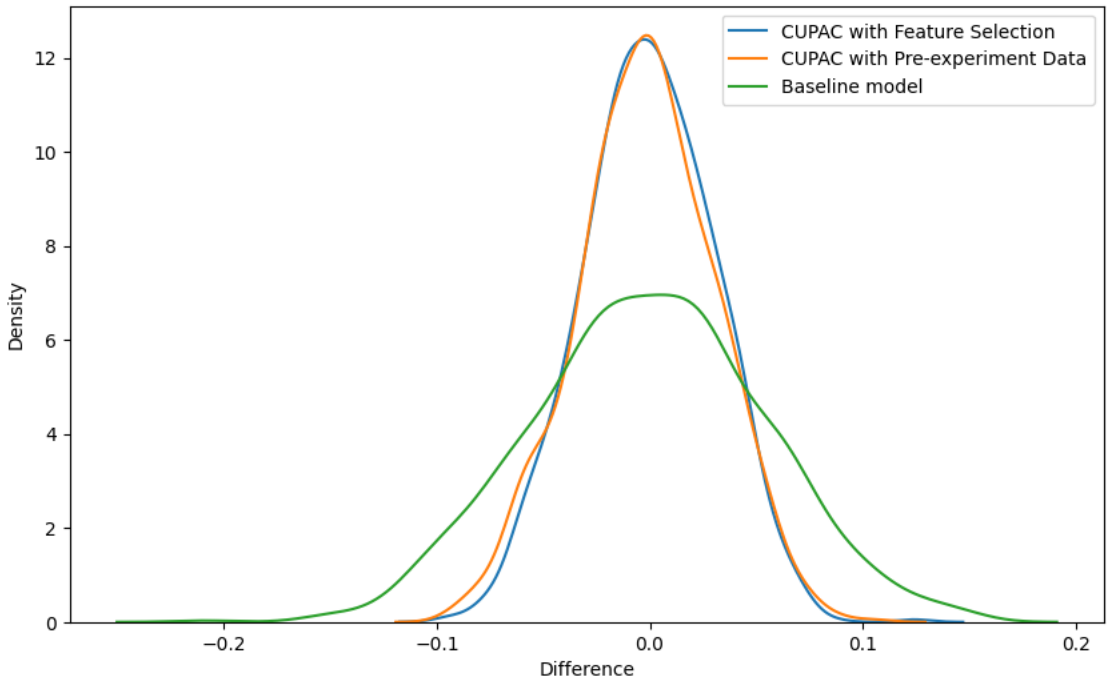
Table 12

Comparison of Critical Metrics of Baseline Model, CUPAC with Feature Selection

	Variance	Variance Reduction Percentage Comparing to the Baseline Model	Mean	Mean Diff Comparing to the Baseline Model
Baseline model	0.002997	0.0%	-0.000953	0.0%
CUPAC with pre-experiment data	0.001041	65.271019%	-0.002028	-112.778525%
CUPAC with feature selection	0.000938	68.693642%	-0.000470	50.700868%

Figure 19

A/A Test Results Comparison – Baseline, CUPAC with Pre-experiment Data, and CUPAC with Feature Selection



5.5 Overall Comparison

Table 13 and Figure 20 record the results of the A/A test for all models. From Table 13, we concluded that all models, except Model 5 (CUPAC without pre-experiment data), successfully reduced the variance of the experimental data. Model 7 (CUPAC with feature selection) reduced the most variance, about 68%, while Model 6 (CUPAC with pre-experiment data) and Model 4 (CUPED with ML) followed with approximately 65% reduction. All models had slightly changed the mean from the baseline. Model 5 (CUPAC without pre-experiment data) has the most minor influence on the mean, which was preferable; Model 4 (CUPED with ML) affected the mean the most, with an 189% shift. The conclusions above were consistent with Figure 20, where the distributions of Model 4, Model 6, and Model 7 were the narrowest, indicating that these models had significant potential to increase sensitivity in a real A/B test. Model 4 was also the most left-skewed, indicating some bias.

CUPAC models like Model 6 and Model 7 performed worse than expected, offering limited improvement compared to Model 3 (Basic CUPED). This result might be due to the limited features available for estimating the covariate. The more correlated the covariate was to the target value, the more variance could be reduced. The regulation for choosing covariates was strict, with most qualified factors being observation-level and pre-experiment data; however, these data types existed limited. Although CUPAC models did not perform as well as expected, they were flexible and had the potential to be extended by experimenting with more factors to estimate the covariate.

On the other hand, Model 3's performance was surprisingly good with such a simple mechanism. The high proportion of available pre-experiment data might be why CUPED is so powerful. For the dataset used in Model 3, 65% of users had pre-experiment data, which was sufficient. In the next section, we tried Model 3 on a new user test, which should provide limited pre-experiment information and checked how Model 3 performs.

Model 3 and Model 7 had the most potential to be developed for optimal variance reduction, considering their high variance reduction rate and more minor mean shifts. Model 4 and Model 6 were also worth considering, given their high reduction rates, although they exhibited a slightly more mean shift.

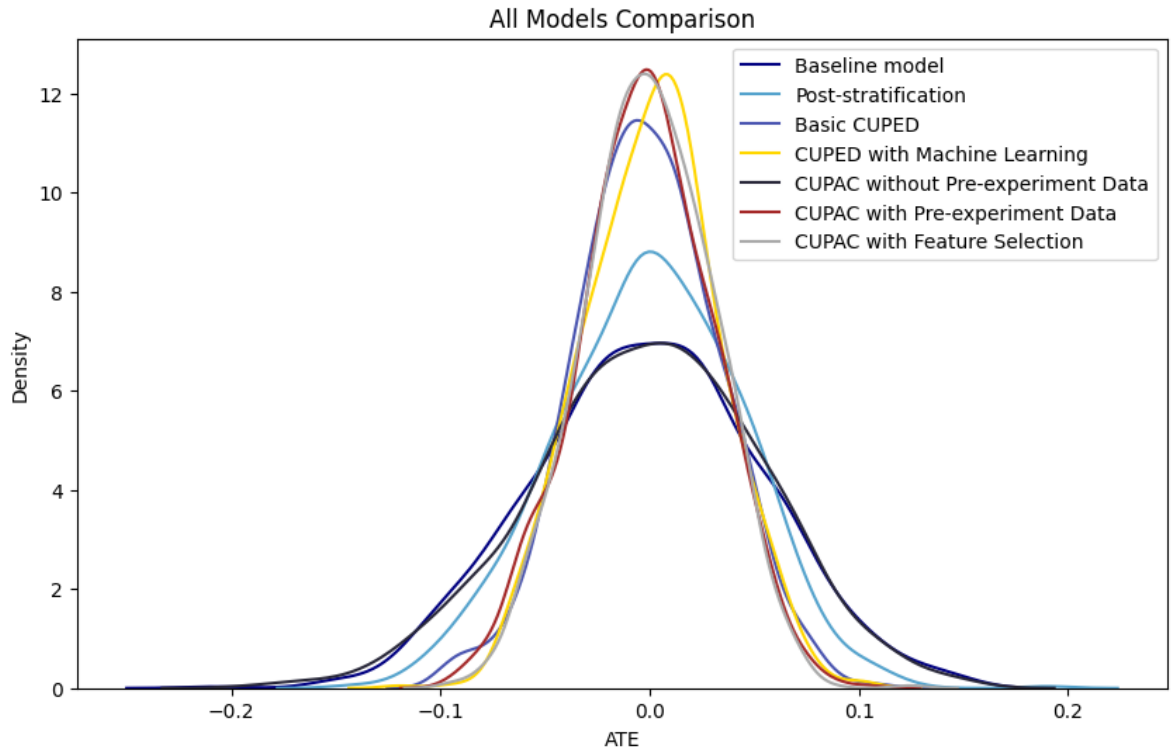
Table 13

Comparison of Critical Metrics of All Models

	Variance	Variance Reduction Percentage Comparing to the Baseline Model	Mean	Mean Diff Comparing to the Baseline Model
Baseline	0.002997	0.0%	-0.000953	0.0%
Post-stratification	0.001997	33.372574%	-0.000463	51.373255%
Basic CUPED	0.001154	61.480803%	- 0.001225	-28.546257%
CUPED with ML	0.001037	65.405063%	0.000850	189.143956%
CUPAC without pre-experiment data	0.003043	-1.542138%	- 0.000866	9.115101%
CUPAC with pre-experiment data	0.001041	65.271019%	-0.002028	-112.778525%
CUPAC with feature selection	0.000938	68.693642%	-0.000470	50.700868%

Figure 20

A/A Test Results Comparison – All Models



Practical Implementation

Based on the simulated A/A test results, we selected four potential models to implement on actual A/B test data recently conducted by Tripledot. The models chosen for further development are Model 3 (Basic CUPED), Model 4 (CUPED with Machine Learning), Model 6 (CUPAC with pre-experiment data), and Model 7 (CUPAC with feature selection). We tested these selected models on two A/B tests, including one test for all users and one test specifically for new users.

The datasets used in these A/B tests were comprehensive and differ from those used in the A/A test section. We meticulously wrote SQL queries to obtain four datasets for each A/B test. Each dataset included all users assigned to the A/B tests, their variant allocation, and all other features used in the models, ensuring a thorough and complete analysis.

The models were adjusted to align with business benefits and address data limitations. First, the number of game starts remained the target metric for evaluating variance reduction performance. However, unlike the machine learning models in the A/A test section, which only considered the number of game starts and its pre-experiment data, the machine learning models in this section were built in a more general way. This approach considered all metrics the company was interested in, making the method more straightforward. For example, in CUPAC models, we built machine learning models to estimate covariates with the number of game starts as the dependent variable and other information, including the pre-experiment number of game starts, as independent variables. In this chapter, machine learning models predicted five critical metrics to get five estimated covariates: the number of game starts, total revenue, number of ad impressions in rewarded windows, number of impressions of interstitial ads, and the number of impressions of banner ads. The independent variables also included the pre-experiment data for all crucial metrics mentioned above. This adjustment aimed to be more aligned with practical conditions since training machine learning models individually for each metric is costly.

The second difference from the previous chapter was the labelling of new or existing users. Instead of labelling new users, we labelled users who were inactive 14 days before being allocated to the experiment. Identifying purely new users who have yet to gain experience playing the game was challenging due to varying user behaviours. We initially labelled new users by comparing their install date to the allocation date, assuming new players were installed and allocated on the same day. However, some users installed the app before the allocation date but just started playing when included in the A/B test. We also tried identifying new users based on missing values for the previous number of game starts and install dates. However, missing game start values did not necessarily indicate that the user had never played. They might have played before the 14-day window or interacted with the app without starting a new game. Therefore, we labelled active users during the 14-day pre-experiment period and used the install cohort day to determine if they had installed the game earlier..

6.1 Evaluate Potential Models on an All-User A/B Test

The first A/B test selected, experiment ID 2686, evaluated a new monetization flow in Solitaire. At Tripledot, we display third-party ads to generate profit. Initially, Solitaire used Max as a third-party provider, but we recently switched to Google, expecting better quality ads and improvements in engagement, retention, and revenue.

A substantial user base of 995,320 was involved in this A/B test, with 82% having pre-experiment play records. The study was divided into four treatment groups, and we randomly selected one, labeled '1.0 Coefficient,' to apply the variance reduction models. We conducted a standard A/B test and then individually implemented the four selected variance reduction tools to compare their results.

Figure 21 shows the 95% confidence interval of the A/B test, comparing treatment and control groups from all models. The four selected models effectively narrowed the confidence interval for both groups, indicating more concentrated distributions. With variance reduction models, treatment

groups generally had higher point estimates than control groups, suggesting a positive effect of the treatments. However, it was still challenging to determine which model performed best in reducing variance and affecting the A/B test results the most.

Figure 21

All User tests, ID 2686, 95% Confidence Absolute Values - Number of Game Start

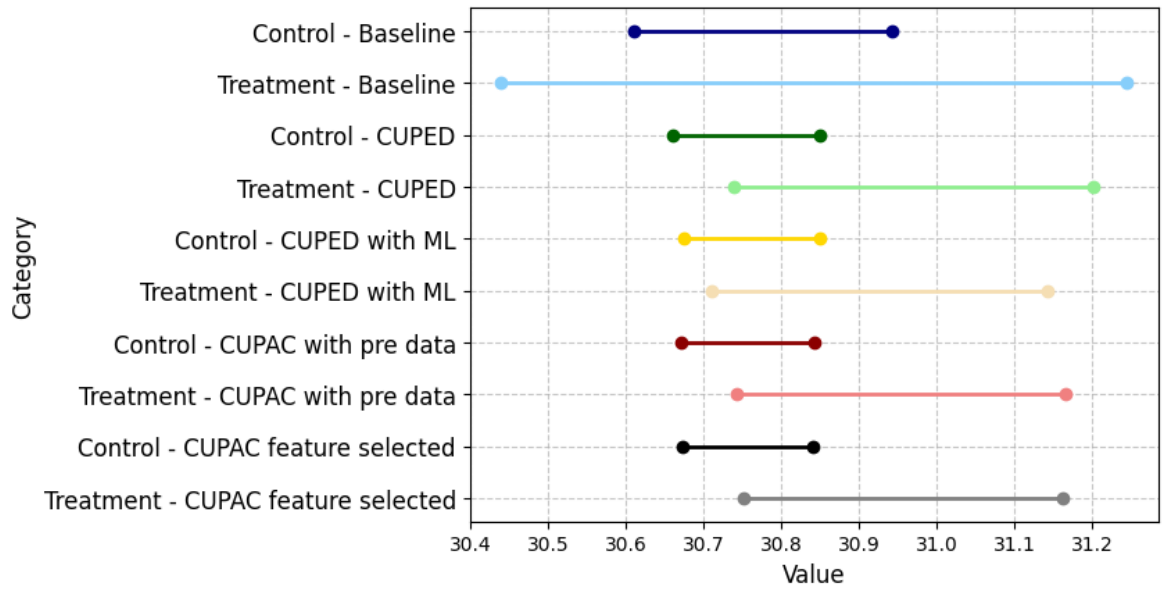


Figure 22 shows how much the confidence interval of the treatment groups is uplifted compared to the control group. One way to interpret the A/B test result is by checking if the treatment uplift crosses 0. If the confidence interval does not cross 0, the A/B test is statistically significant, indicating that the treatment group differs from the control group. In Figure 22, the baseline interval ranges from slightly negative to slightly positive, showing no expected uplift or decrease in performance.

However, the uplifts of Model 3 (Basic CUPED), Model 6 (CUPAC with pre-experiment data), and Model 7 (CUPAC with feature selection) do not cross 0, indicating significant A/B test results after implementing these models. Model 4 (CUPED with machine learning) does not perform as well, possibly due to its left-skewed issue.

Figure 22

All User tests, ID 2686, 95% Confidence Intervals of Treatment Uplift - Number of Game Start

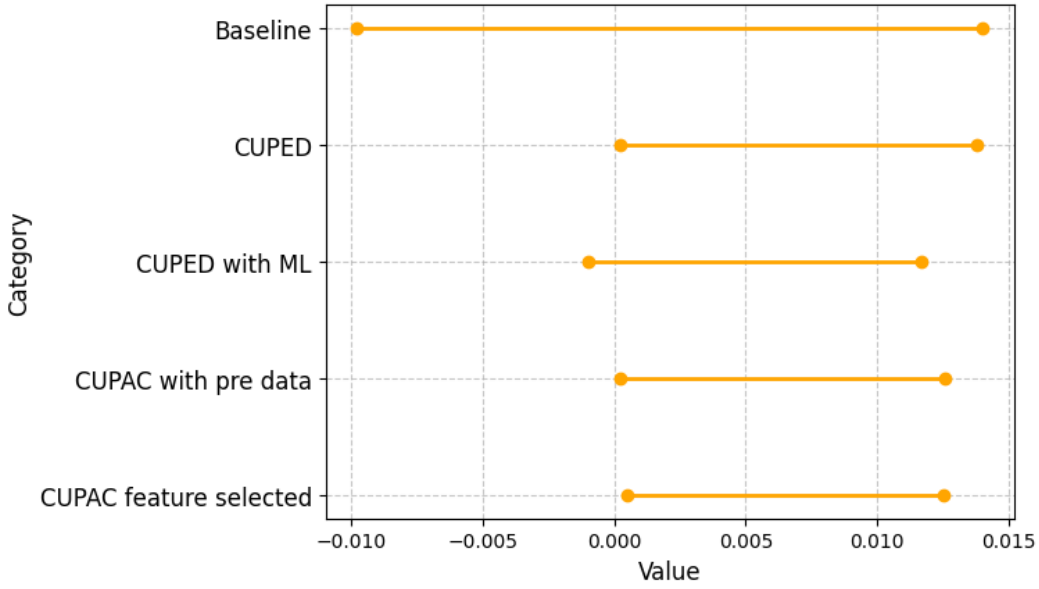


Table 14 shows the A/B test results for each technique. All models successfully reduced the variance of the treatment group. Model 7 (CUPAC with feature selection) reduced variance the most, by 73.8%, followed by Model 6 (CUPAC with pre-experiment data) at 72.3%, Model 4 (CUPED with ML) at 71.1%, and Model 3 (Basic CUPED) at 66.9%. Model 7 only slightly outperformed Model 3 because the high proportion of existing users with pre-experiment data provided sufficient information for CUPED.

Initially, the A/B test result was insignificant, with a baseline p-value of 0.38. However, with reduced variance, the results of Models 3, 6, and 7 became statistically significant, with p-values below 0.05. Despite substantial variance reduction, Model 4's result remained insignificant, affected by variance and effect size. Table 14 shows that Model 4 (CUPED with ML) had an effect size of 0.16, lower than Model 3's 0.21, despite having a more minor variance. This suggests instability in Model 4's machine learning approach, possibly due to model complexity, overfitting, and adjustments diluting the treatment effect, leading to an insignificant result even with low variance.

Another noteworthy observation was that even though Model 7 (CUPAC with feature selection) had a smaller effect size than Model 3 (Basic CUPED), its p-value was minor, indicating that the result was more significant than others. This phenomenon could be explained by the fact that the p-value was influenced by both the effect size and the standard error, a function of variance and sample size. Model 7 employed an improved variance reduction strategy, leading to a smaller p-value despite a slightly smaller effect size.

To sum up, Model 7 outperformed all other techniques, producing the smallest p-value and the narrowest distribution. The result demonstrated its superior ability to detect small effects in an A/B test.

Table 14

Comparison of Critical Metrics of All Models on an All-User A/B test

	P-value	Treatment Variance	Treatment Variance Reduction Comparing to the Baseline Model	Effect Size
Baseline	0.384538	4217.5674	0.0%	0.065210
Basic CUPED	0.046055	1394.5660	66.934351%	0.214753
CUPED with ML	0.084862	1216.1989	71.163496%	0.163962
CUPAC with pre-experiment data	0.045515	1164.8754	72.380396%	0.201657
CUPAC with feature selection	0.038852	1101.4462	73.884323%	0.194400

Note. The effect size is the difference of treatment and control groups in mean.

6.2 Evaluate Potential Models on a New-User A/B Test

The selected new-user A/B test, experiment 2660, aimed to assess the effectiveness of a new animation after winning a game. The animation was designed to make the celebration of winning more exciting. The product team expected that introducing this new animation in the treatment group would increase the initiation of new games. 621,100 new users were allocated to this experiment,

with 310,030 in the treatment group and 311,070 in the control group. Since this was a new-user test, only 1.76% of users had pre-experiment data, providing limited information to control for covariates.

Figure 23 visualizes the 95% confidence intervals for the absolute number of game starts across different models and variants. Compared to Figure 21, all confidence intervals were barely narrowed. Model 3 (Basic CUPED) intervals were nearly identical to the baseline, showing minimal influence. Model 4 (CUPED with ML) and Model 7 (CUPAC with feature selection) had the narrowest intervals, providing the most consistent and precise estimates.

Figure 24 represents the uplift value, the relative change in-game starts due to the treatment compared to the control. Each horizontal line shows the 95% confidence interval for the uplift value for each method. All intervals included 0, indicating no statistically significant uplift. This suggests that the treatments did not have a reliably positive or negative impact on the number of game starts compared to the control. However, the intervals for Model 4 (CUPED with ML), Model 6 (CUPAC with pre-experiment data), and Model 7 (CUPAC with feature selection) were narrower, though the changes were too limited to be easily discernible from the graph.

Figure 23

New User test, ID 2660, 95% Confidence Absolute Values - Number of Game Start

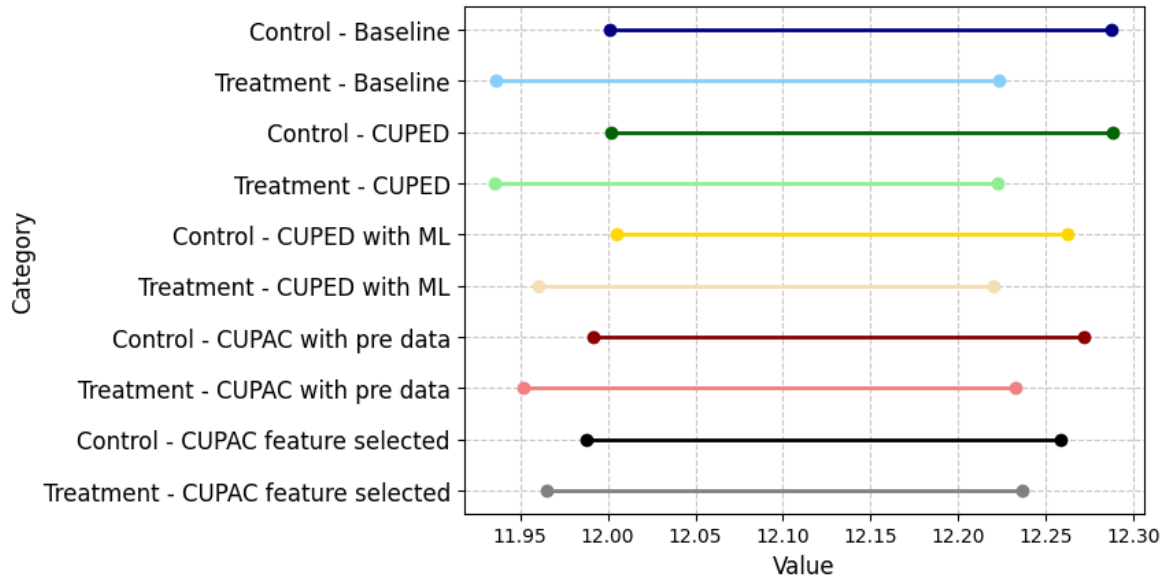


Figure 24

New User test, ID 2660, 95% Confidence Intervals of Treatment Uplift - Number of Game Start

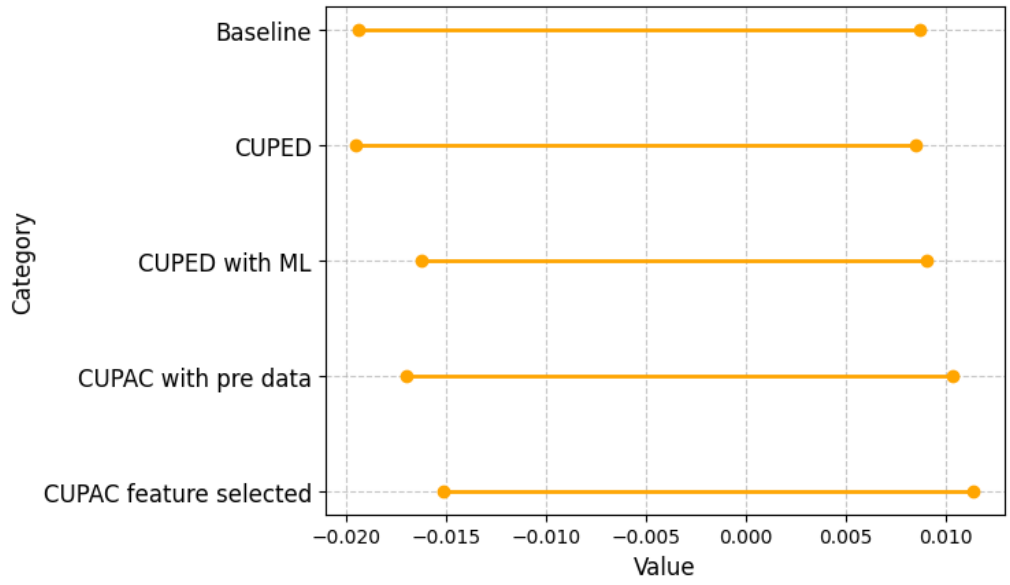


Table 15 aligns with Figure 24, showing that none of the models had a p-value below 0.05, indicating no statistically significant change in the number of game starts. Regarding variance, Model 4 (CUPED with ML) achieved the highest reduction (18.36%), followed by Model 7 (CUPAC with feature selection) at 10.11%, suggesting these methods were most effective at reducing variance. The effect

sizes for all models were negative, indicating that treatment groups generally had fewer game starts than control groups. However, the CUPAC feature selection model showed the least negative impact.

In conclusion, Model 7 (CUPAC with feature selection) handled the new user test best, providing consistent results with minor variance and the smallest adverse effect size. Model 3 (Basic CUPED) barely functioned effectively in a new user test due to the lack of pre-experiment data. Model 4 (CUPED with ML) also reduced variance but significantly affected the treatment estimate.

Table 15

Comparison of Critical Metrics of All Models on a New-User A/B test

	P-value	Treatment Variance	Treatment Variance Reduction Comparing to the Baseline Model	Effect Size
Baseline	0.735250	1662.063535	0.0%	-0.065031
Basic CUPED	0.739621	1659.534360	0.152170%	-0.066366
CUPED with ML	0.679488	1356.940694	18.358073%	-0.043448
CUPAC with pre-experiment data	0.653971	1592.617736	4.178287%	-0.040077
CUPAC with feature selection	0.590676	1494.066969	10.107710%	-0.022469

Note. The effect size is the difference of treatment and control groups in mean.

Conclusion

This chapter will summarize all model results and provide business recommendations. We will also provide some suggestions for extending the research in the future.

7.1 Summary and Business Recommendation

This project aims to develop variance reduction techniques to enhance the accuracy of A/B tests conducted within the company, thereby strengthening data-driven decision-making. This paper evaluates three methods and seven models by running simulated A/A tests and implementing potential models on actual A/B test data.

Based on the A/A test results, Model 7 (CUPAC with feature selection), Model 6 (CUPAC with pre-experiment data), Model 4 (CUPED with ML), and Model 3 (Basic CUPED) reduce variance by approximately 60% to 68%, which can significantly impact the experiment results. However, we also find that Model 4 (CUPED with ML) introduces slight bias, possibly due to the machine learning model used in CUPED.

We then applied the four models mentioned above to actual A/B tests, including an all-user test and a new-user test. Model 7 (CUPAC with feature selection) performs the best in both tests, providing the smallest p-value by reducing variance and precisely estimating the effect size. Model 4 (CUPED with ML) is unstable enough for company-wide implementation. Although Model 4 significantly reduces variances, it also reduces the effect size, affecting the p-value and the model's robustness. Model 3 (Basic CUPED) performs well on the all-user test but has minimal impact on the new-user test due to the lack of pre-experiment data to control for covariates.

We recommend developing Model 7 (CUPAC with feature selection) at Tripledot for implementation in various A/B tests. Model 7 demonstrates stable performance in both simulated A/A

and A/B tests and handles all user tests effectively. Additionally, Model 7 has the potential to extend more features to estimate covariates and to be built with different machine-learning models, allowing room for further improvement. However, considering the data pipeline resources required for building a variance reduction tool, Model 3 (Basic CUPED) is preferable due to its simple algorithm and data query requirements. Model 3 performs slightly worse in all-user tests than Model 7 but still provides sufficient power to optimize the experiment. Although Model 3 has little influence on the new-user test, this drawback can be overlooked when considering resource savings. A more complex model like Model 7 only reduces variance by a limited 5% in the new-user test, indicating that all models perform poorly in handling new-user tests.

7.2 Limitations and Future Work

In building all statistical models for the A/A test section, we encountered a significant lack of code resources, research, and practical information, particularly concerning CUPAC and improved CUPED techniques, such as Model 4 (CUPED with ML). This scarcity made the process both challenging and time-consuming and limited the potential for further enhancement of individual models.

The CUPAC model holds significant potential, with many aspects available for improvement, such as the selected features and the machine learning model used to estimate covariates. By identifying more suitable features, the CUPAC model, particularly Model 7 (CUPAC with feature selection), might enhance its ability to handle new-user A/B tests—something that CUPED has not been able to achieve.

References

Bonet, D. M. (2023, January 10). *Variance reduction in experiments using covariate adjustment techniques.* Medium. https://medium.com/glovo-engineering/variance-reduction-in-experiments-using-covariate-adjustment-techniques-717b1e450185

Boyle, P. P. (1977). Options: A Monte Carlo approach. *Journal of Financial Economics, 4*(3), 323–338. https://doi.org/10.1016/0304-405X(77)90005-8

Cochran, W. G. (1977). *Sampling techniques.* John Wiley & Sons.

Corotto, F. S. (2022). *Wise Use of Null Hypothesis Tests: A Practitioner's Handbook (1st ed.).* Elsevier Science & Technology. https://doi.org/10.1016/C2021-0-02572-4

Deng, A., Xu, Y., Kohavi, R., & Walker, T. (2013). *Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data.* https://exp-platform.com/Documents/2013-02-CUPED-ImprovingSensitivityOfControlledExperiments.pdf

Glasgow, G. (2005). Stratified sampling types. In Elsevier eBooks (pp. 683–688). https://doi.org/10.1016/b0-12-369398-5/00066-9

Gupta, S., Kohavi, R., Tang, D., Xu, Y., Andersen, R., Bakshy, E., Cardin, N., Chandran, S., Chen, N., Coey, D., Curtis, M., Deng, A., Duan, W., Forbes, P., Frasca, B., Guy, T., Imbens, G. W., Saint Jacques, G., Kantawala, P., & Katsev, I. (2019). Top Challenges from the First Practical Online Controlled Experiments Summit. *ACM SIGKDD Explorations Newsletter*, *21*(1), 20–35. https://doi.org/10.1145/3331651.3331655

Holt, D., & Smith, T. M. F. (1979). Post Stratification. *Journal of the Royal Statistical Society. Series A. General, 142*(1), 33–46. https://doi.org/10.2307/2344652

Jackson, S. (2018, Jan 22). *How Booking.com increases the power of online experiments with CUPED.* Medium. https://booking.ai/how-booking-com-increases-the-power-of-online-experiments-with-cuped-995d186fff1d

Jager, K. J., Zoccali, C., MacLeod, A., & Dekker, F. W. (2008). Confounding: What it is and how to deal with it. *Kidney International, 73*(3), 256–260. https://doi.org/10.1038/sj.ki.5002650

Keppel, G., Saufley, W. H., & Tokunaga, H. (1998). *Introduction to design and analysis: a student's handbook.* W.H. Freeman.

Kevin R. Murphy, & Brett Myors. (2023). *Statistical Power Analysis, 5th Edition.* Routledge.

Kilby, N. (2023, March 1). FT 1000: the seventh annual ranking of Europe's fastest-growing companies. Financial Times. https://www.ft.com/ft1000-2023

Kohavi, R., Tang, D., & Ya Xu. (2020). *Trustworthy online controlled experiments: a practical guide to A/B testing.* Cambridge University Press.

Tang, Y., Caixia Huang, David Kastelman, & Jared Bauman. (2020). Control using predictions as covariates in switchback experiments. *DoorDash Inc.* https://doi.org/10.13140/RG.2.2.34500.04488

Tripledot Studios. (2024). Tripledot Studios. Tripledot Studios. https://tripledotstudios.com/

Rubinstein, R. Y., & Marcus, R. (1985). Efficiency of Multivariate Control Variates in Monte Carlo Simulation. *Operations Research, 33*(3), 661–677. http://www.jstor.org/stable/170564

Valliant, R. (1993). Poststratification and Conditional Variance Estimation. *Journal of the American Statistical Association, 88*(421), 89–96. https://doi.org/10.2307/2290701

Yang, S. (2021, September 13). *Online Experiments Tricks — variance reduction - towards data science.* Medium. https://towardsdatascience.com/online-experiments-tricks-variance-reduction-291b6032dcd7

Xie, H., & Aurisset, J. (2016). Improving the Sensitivity of Online Controlled Experiments: Case Studies at Netflix. P*roceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 645–654. https://doi.org/10.1145/2939672.2939733

Appendices

I. Code Repository

All code and models can be found in the GitHub repository:

https://github.com/wangpoyu/UCL_dissertation_variance_reduction_abtest.git

II. Project Management

We organized the project using the Kanban board in the Ai8 project management system. In this section, we provide the structure of all tasks and screenshots of the board every two weeks since the project started.

Overall, the structure of all tasks is shown as below:

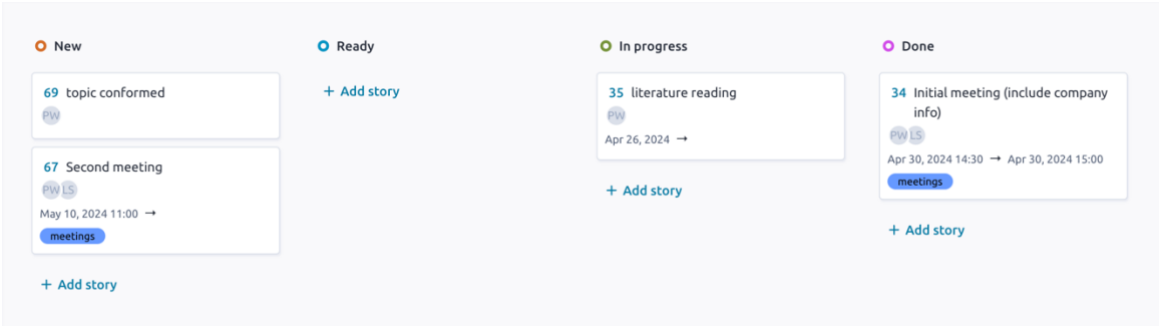
1. Initial set up work
    a. Literature reading
    b. Topic confirmed
    c. Techniques selection
    d. Query data from database
2. Code up A/A test and models
    a. Build baseline model
    b. Build post-stratification mode
    c. Build CUPED models
    d. Build CUPAC models
    e. Improve CUPAC models
3. Implement models on actual A/B test
    a. Implementation on an all-user test
    b. Implementation on a new-user test
4. Writing
    a. Structure the dissertation outline

b. Literature review writing
c. Methodology writing
d. A/A test results writing – baseline
e. A/A test results writing – post-stratification
f. A/A test results writing – CUPED models
g. A/A test results writing – CUPAC models
h. Practical implementation writing – all-user test
i. Practical implementation writing – new-user test
j. Conclusion
k. Introduction

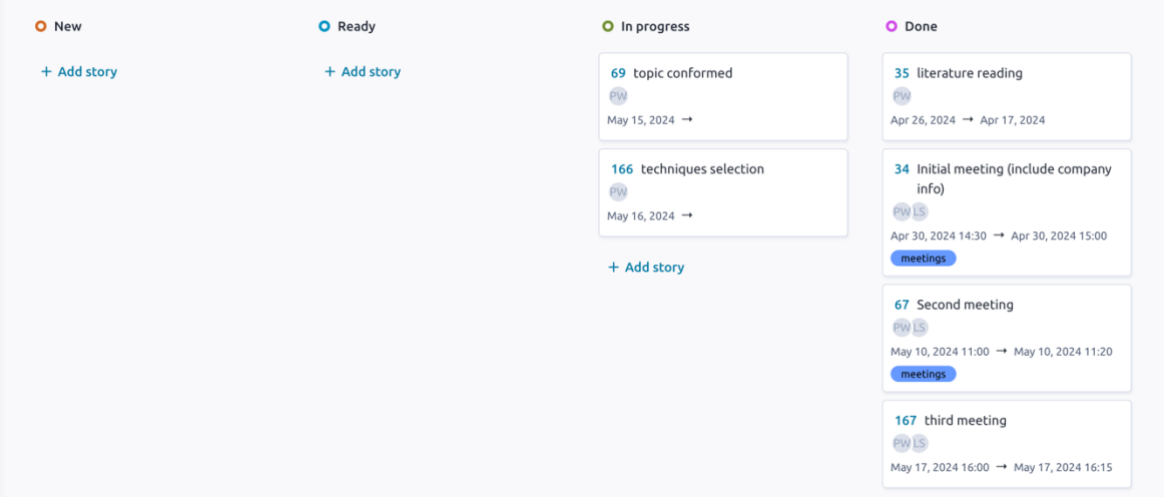
5. Final organises
    a. Formatting
    b. Organise code files

The screenshots of Kanban board are shown as bellow:

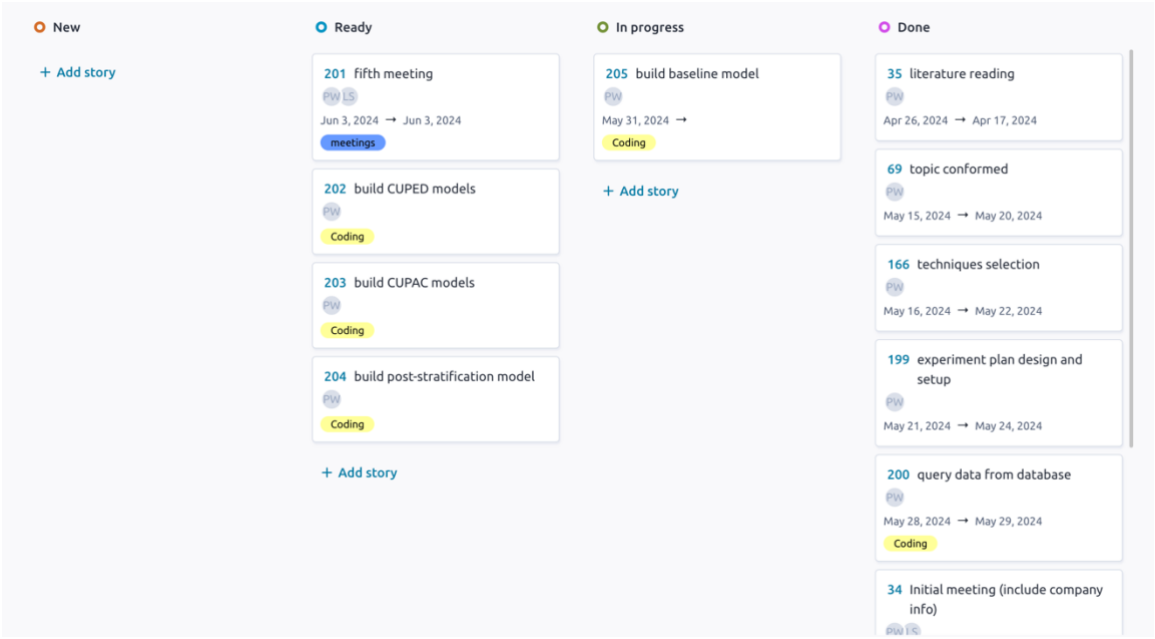
1. Screenshot on 5th May 2024



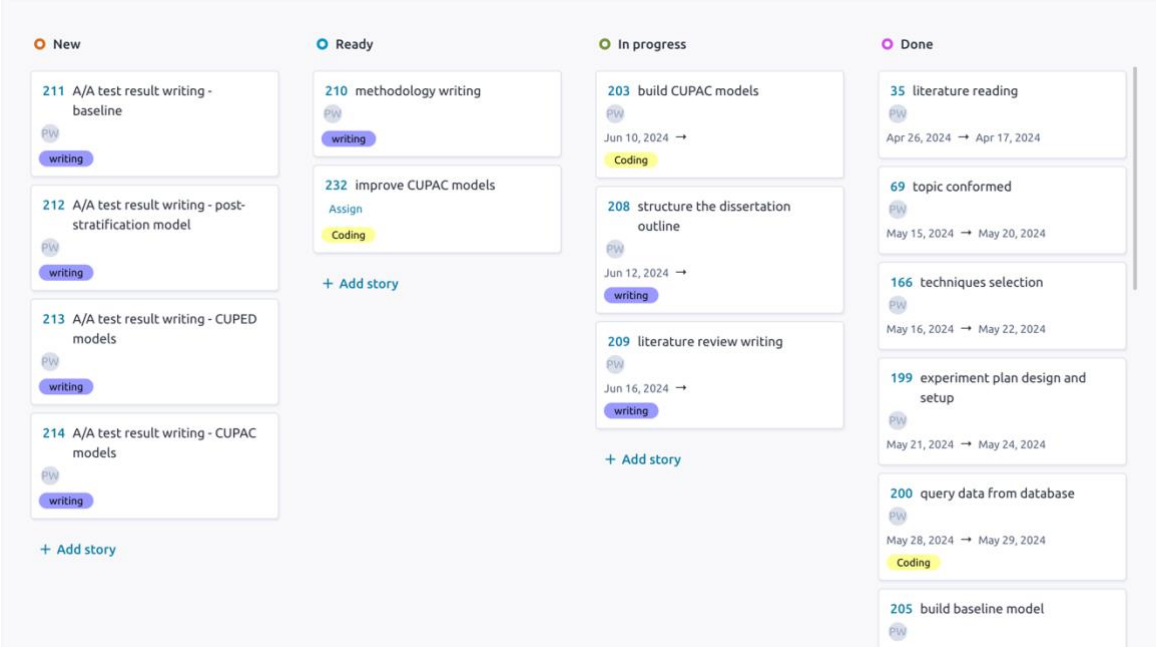
2. Screenshot on 19th May 2024



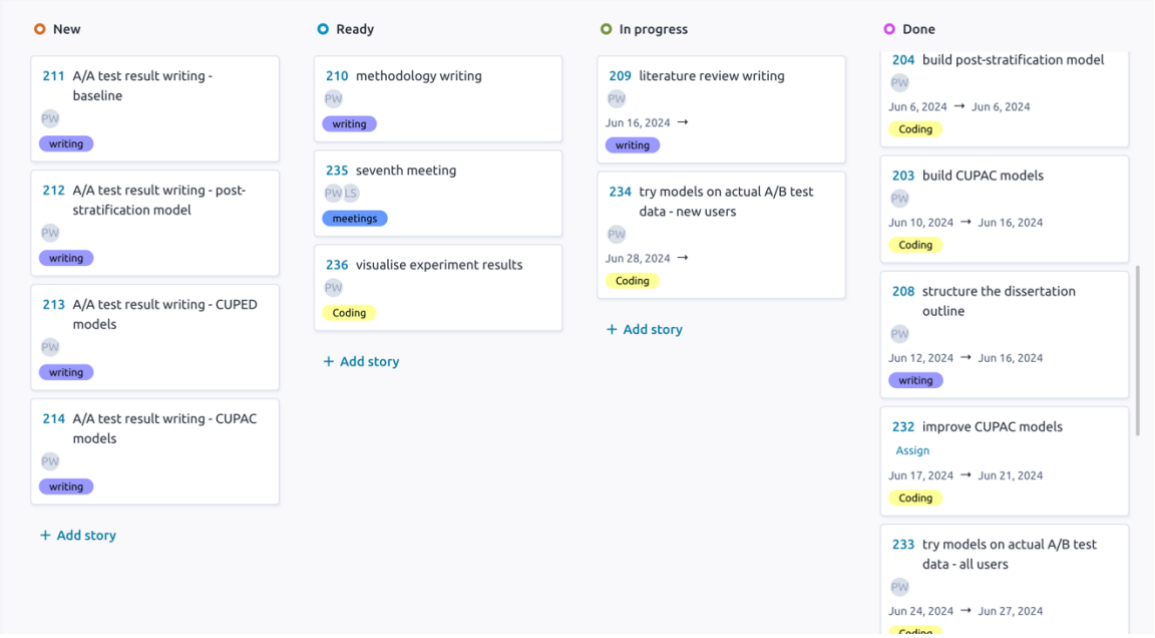
3. Screenshot on 2nd June 2024



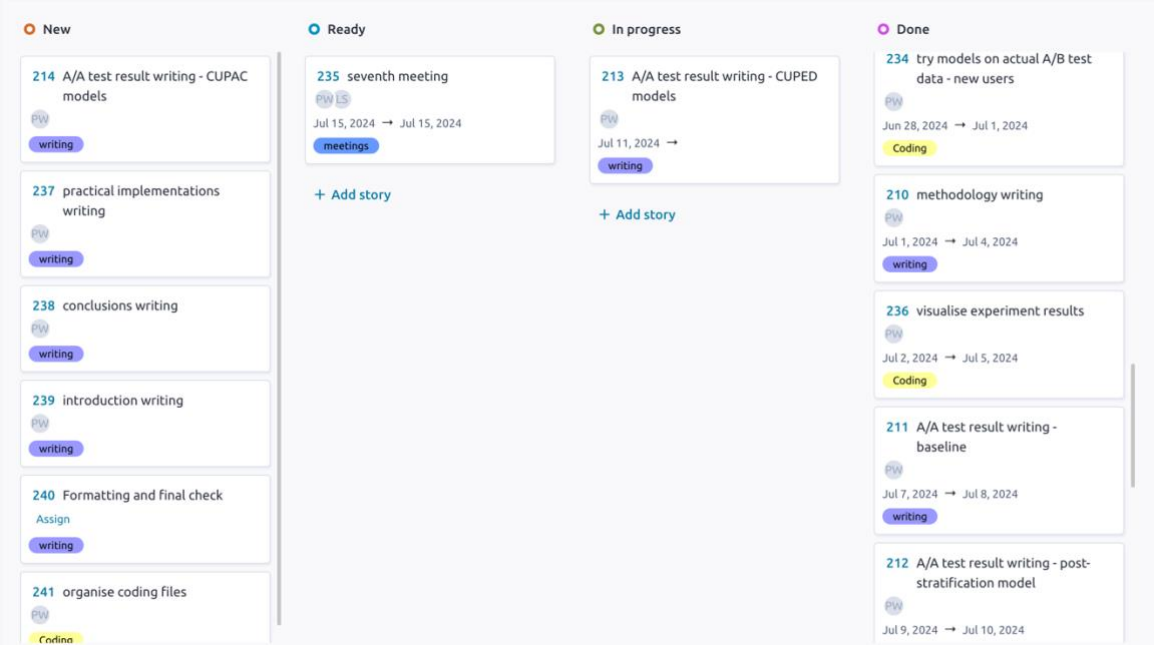
4. Screenshot on 16th June 2024



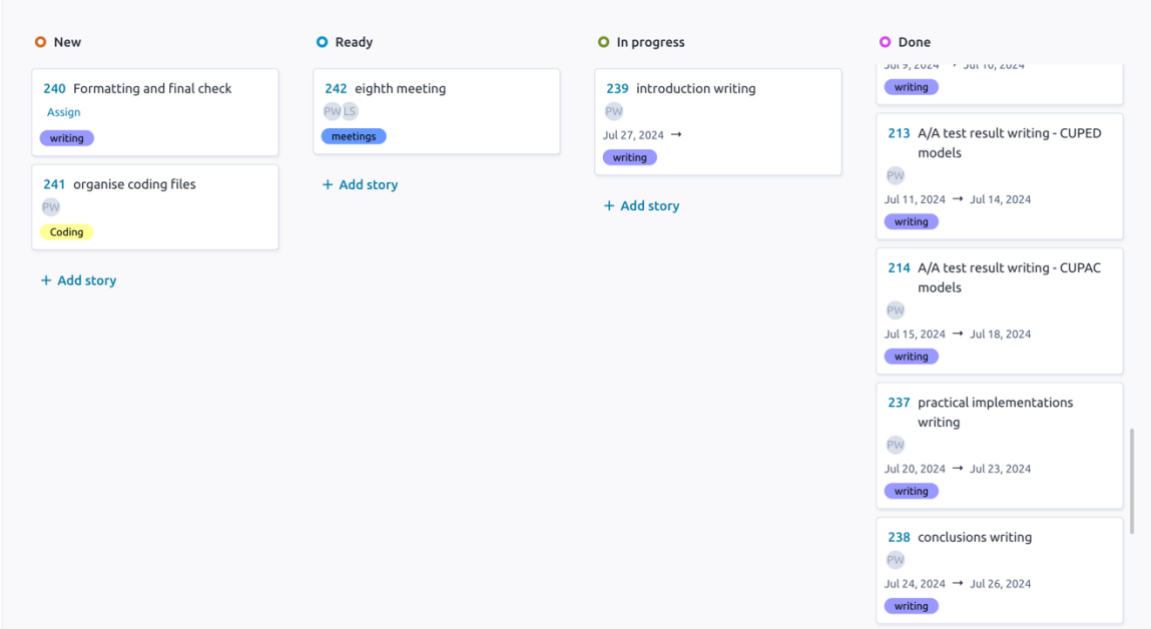
5. Screenshot on 30th June 2024



6. Screenshot on 14th July 2024



7. Screenshot on 28th July 2024



8. Screenshot on 3rd July 2024