Index

1. Executive Summary

This report presents an analysis of the experimental data from an A/B test conducted within the Loan Review Department of a consumer lending company. The experiment aimed to assess whether a newly developed computer model improves loan officers' decision-making accuracy compared to the existing model.

2. Introduction

The company processes loan applications ranging from $20,000 to $35,000. Its primary objective is to minimise financial losses from defaulted loans while maximising profits from successful repayments. Loan approval decisions are made by loan officers with the assistance of a computer model that predicts whether a loan should be approved or rejected.

Recently, the company has experienced high error rates in loan approvals, leading to significant financial losses. To address this issue, a new computer model has been developed to improve decision-making. The task is to analyse the results of a small-scale A/B test conducted within the Loan Review Department. This experiment compares the performance of loan officers using the existing model versus those using the new model.

This report will evaluate the effectiveness of the new model by analysing key decision metrics. Based on the findings, recommendations are provided on whether the new model should be implemented, if further testing is needed, or if modifications to the experiment are required.

3. Experiment Design and Data Collection

3.1. A/B testing

A/B Testing is a statistical method for comparing two alternatives of a variable in terms of performance and selecting one that works best (Kohavi et al., 2009). The variable is divided into two groups, Control and Treatment.

Control Group: Participants who continue using the existing system or method, serving as a baseline for comparison. In this case, loan officers continued using the existing computer model for loan approvals.

Treatment Group: Changes in the existing model to assess its effectiveness. In this case, Loan officers used the new computer model, designed to improve decision accuracy.

3.2. Randomisation Unit

In this experiment, the randomisation unit was the loan officer, meaning that each loan officer was individually and independently assigned to either the control or treatment group. This randomised controlled trial ensures that the comparison is unbiased and that differences in performance are attributable to the new model rather than individual skill variations.

4. Data Preparation and Analysis Methodology

4.1. Data Preparation

R studio was used to check for duplicates, data structure, outliers, misspellings, and accuracy. The analysis confirmed the dataset is clean with no significant issues (Appendix A for details). However, four key observations were noted upon review

1. **Experiment Imbalance:** Unequal numbers of experiments between 190 control and 280 treatment groups.

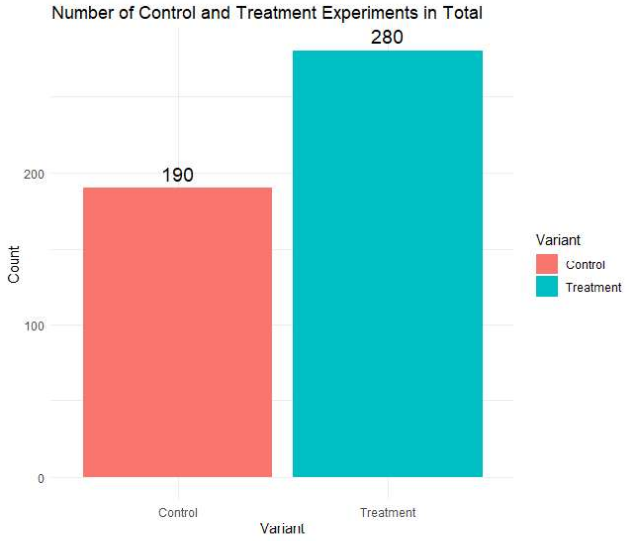


Figure 1: Number of Control and Treatment Experiments in Total

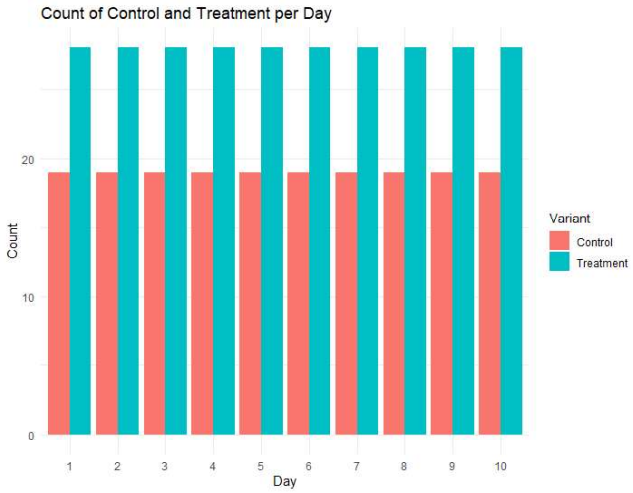


Figure 2: Count of Control and Treatment Per Day

2. **Decisions Without Model:** Some staff bypassed decision-making in the final stage, as indicated by cases where they made initial-stage decisions but recorded 0 final-stage decisions. (complt_init > 0 but complt_fin = 0). Additionally, all confidence scores for these cases were 0, reinforcing this assumption.
3. **Decision Count Discrepancies:** Mismatches in agreed and conflict decisions were detected in 90 control experiments by comparing total counts across decision stages.
4. **Total Loan Decisions:** Some control experiments showed inconsistencies in daily loan decisions, which were expected to total 10 per day. These were checked using summary statistics on the number of loans in the final stage (complt_fin) per experiment.

To evaluate the OECs and other metrics, 90 incomplete control experiments were removed. Since OEC measures the model's impact on loan reviews, only model-based decisions are analysed. This leaves 100 control and 280 treatment experiments, which may limit the reliability of results. (Appendix B for further information)

4.2. Overall Evaluation Criterion (OEC)

The Overall Evaluation Criterion (OEC), a predefined objective used to assess an experiment's impact. It serves as a performance benchmark to determine whether the treatment group performs significantly better than the control group.

Given the company's objective, the OEC in this analysis is the Type II error rate, as minimising the approval of bad loans is the primary goal.

4.3. Metrics Used for Analysis

1. **Error in the decision in final Stage**
    a. **Type I Error**: The number of loans where loan officers have rejected good loans.
    b. **Type II Error**: The number of loans where loan officers have approved bad loans.
2. **Agreement Metrics**
    a. The proportion of loan officers' agreements with computer predictions after seeing the model predictions.
3. **Decision Revision Metrics**
    a. The proportion of decisions revised by loan officers to align with computer model predictions.

4. **Confidence Score**
    a. Loan officers' decision confidence score after seeing the computer model's predictions.

While reducing bad loan approvals is critical, practical adoption by loan officers is equally important. A model, despite strong performance, may not be adopted if staff lack confidence or trust in it. Therefore, staff confidence and alignment with model decisions are also considered (Choudhary, 2022).

4.4. Analysis Methodology

A statistical approach is employed to ensure the reliability and validity of the findings. The following methods are used:

1. Hypothesis Testing: Determines whether differences in the OEC and other metrics between the old and new models are statistically significant.
2. Confidence Intervals: Quantifies the uncertainty around estimated metrics and assesses result reliability.
3. Welch's two-sample t-tests: Compares the means of key metrics between the old and new models.
4. Effect Size (Cohen's d): Quantifies the magnitude of differences between the old and new models, indicating practical significance

4.5. Results and Interpretation

During the EDA, it was identified that not all loan applications were reviewed using the model. This means that loan officers could evaluate clients independently, avoiding the model’s predictions. This scenario may affect the A/B testing results since instances, where the model was not utilised, are unlikely to influence the difference between the treatment and control groups in our OEC: Type II error. Therefore, two scenarios were considered: Filtered Data (excluding cases where no loan review decisions were made after seeing the model's predictions) and Whole Data (including all cases).

4.5.1. OEC: Type II error

$$Type\ II\ error = \frac{False\ Negatives}{Total\ Actual\ Positives} = \frac{\text{typeII\_fin}}{\text{badloans\_num}}$$

A t-test was conducted to measure whether the new model reduces the type II error.

Hypothesis:

Null: No difference in Type II error between Control and Treatment groups.

Alternative: The Treatment group has less Type II Error Rate (one-tailed t-test).

Results:

Scenario	P-value t-test	Mean Control	Mean Treatment	Cohen's d	% Difference (Treatment - Control)
Whole data	0.245	0.207	0.267	0.42 (Small effect)	29.0%
Filtered data	0.005	0.393	0.267	-1.67 (Large effect)	-31.9%

Table 1: Type II error

In the whole data scenario, the new model did not significantly reduce Type II error compared to the existing model (p = 0.245, d = 0.42).

In the filtered data scenario, the new model significantly reduced Type II error compared to the existing model (p = 0.005, d = -1.67) by 31.9%. This indicates a statistically and practically significant improvement in reducing Type II errors.

Only one scenario (filtered data) was evaluated for the other secondary metrics.

4.5.2. Metric 1: Type I error

$$Type\ I\ error = \frac{False\ Positives}{Total\ Actual\ Negatives} = \frac{\text{typeI\_fin}}{\text{goodloans\_num}}$$

A t-test was conducted to measure whether the new model reduces the type I error.

Hypothesis:

Null: No difference in Type I error between Control and Treatment groups.

Alternative: The Treatment group has less Type I Error Rate (one-tailed t-test).

Results:

Variable	P-value t-test	Mean Control	Mean Treatment	Cohen's d	% Difference (Treatment - Control)
Type I error	0.001	0.502	0.277	-1.96 (Large effect)	-44.9%

Table 2: Type I error

The results show that the new model significantly reduced Type I error compared to the existing model (p = 0.001, d = -1.96) by 44.9%, demonstrating both statistical and practical significance.

4.5.3. Metric 2: Loan Officer Agreement with Model (agree_ratio):

A t-test assessed whether the new model improves loan officers' agreement with model predictions, using agree_ratio (agree_fin/complt_fin)

Hypothesis:

Null: No difference in agreement ratio between the Control and Treatment groups.

Alternative: The Treatment group has a higher agreement ratio with model.

Results:

Variable	P-value t-test	Mean Control	Mean Treatment	Cohen's d
agree_ratio	0.005333	0.7491300	0.8724951	-1.6 (Large effect)

Table 3: Loan Officer Agreement with model (agree_ratio)

Since p-value < 0.05, the null hypothesis is rejected, indicating that Treatment loan officers align more with model prediction. The large effect size (d = -1.6) suggests increased trust or improved model predictions, potentially enhancing decision accuracy and reducing financial risk.

4.5.4. Metric 3: Loan Officer Final Confidence (confidence_fin_total):

A t-test analysed whether model exposure increases loan officers' confidence, using their total final-decision confidence score.

Hypothesis:

Null: No difference in final confidence scores between Control and Treatment groups.

Alternative: The Treatment group has higher final confidence.

Results:

Variable	P-value t-test	Mean Control	Mean Treatment	Cohen's d
confidence_fin_total	4.687e-09	595.1200	730.6286	-0.63 (Medium effect)

Table 4: Loan Officer Final Confidence (confidence_fin_total)

Since p-value < 0.001, the null hypothesis is rejected, indicating Treatment loan officers report significantly higher confidence. The effect size (d = 0.63) suggests a moderate-to-large impact, potentially improving decision-making, efficiency, and reducing decision fatigue, leading to better loan approval accuracy and overall performance.

4.5.5. Metric 4: Loan Officer Decision Revisions (Revised ratio)

A t-test analysed whether the new model increases decision revisions based on model prediction, using revised_ratio (revised_per_ai / complt_fin).

Hypothesis:

Null: No difference in decision revisions between the Control and Treatment groups.

Alternative: The Treatment group has a higher proportion of revised decisions.

Results:

Variable	P-value t-test	Mean Control	Mean Treatment	Cohen's d
conf_change	0.005371	0.05545891	0.12020869	-0.77 (Medium effect)

Table 5: Loan Officer Decision Revisions (Revised ratio)

Since p < 0.01, we reject the null hypothesis, concluding that loan officers in the Treatment group revised significantly more decisions following model prediction. Cohen's d = 0.77 indicates a moderate effect, suggesting that the new model has a meaningful impact on decision revision rates, potentially increasing alignment with model prediction.

5. Business Implications

The new model improves decision accuracy by reducing both Type I and Type II errors, leading to fewer bad loans and more approved good loans—reducing losses and increasing revenue. Higher model adoption and agreement with recommendations suggest loan officers trust the model, improving efficiency and reducing hesitation.

6. Recommendations

1. Extend Experiment Duration: Continue the experiment for at least 6-8 weeks to gather more data and ensure conclusive results. Stopping now could lead to biased or incomplete findings
2. Ensure Adequate Sample Size: The estimated required sample size per variant is 400 (based on Cohen's d = 0.2), ensuring 80% statistical power at a 0.05 significance level.
3. Prevent Model Bypassing: Implement stricter enforcement to ensure that all loan officers use the model, preventing behaviours that could skew results.
4. Measure Operational Efficiency: Track decision-making time to evaluate whether the model improves efficiency and reduces cognitive load for loan officers.
5. Phased Rollout: Expand the experiment through a gradual rollout across the organisation to validate the model's effectiveness in real-world settings.
6. Training for Loan Officers: Provide training programs to improve loan officers' confidence in the model, encouraging adoption and ensuring effective decision-making.

7. Conclusion

The results indicate that the new model significantly reduces both Type I and Type II errors, leading to fewer incorrect loan approvals and rejections. Additionally, loan officers using the new model demonstrated a higher level of agreement with predictions and reported increased confidence in their decisions, suggesting greater trust in the system.

To ensure more reliable results, the experiment should continue with an improved design. The control and treatment groups should be balanced, and all loan officers must complete final-stage decisions to maintain consistency. Additionally, a larger dataset is necessary to enhance the statistical power of the experiment. Strict enforcement of model usage will also be required to accurately measure its impact.

8. References

Kohavi, R., Longbotham, R., Sommerfield, D. and Henne, R.M. (2009). Controlled Experiments on the web: Survey and Practical Guide. *Data Mining and Knowledge Discovery*, 18(1), pp.140–181. https://ai.stanford.edu/~ronnyk/2009controlledExperimentsOnTheWebSurvey.pdf

Choudhary, S.R. (2022). *The No Jargon Guide to Understanding A/B Testing Metrics*. [online] A/B Testing Software. Available at: https://www.convert.com/blog/a-b-testing/ab-testing-metrics-guide/ [Accessed 9 Feb. 2025].

9. Appendices

Appendix A: Exploratory data analysis (EDA)

```
#Library
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(readr)
library(effectsize)
library(pwr)
library(ggplot2)
```

Appendix A1: Data Quality Check

```
# Read the data
LoanData <- read_csv("ADAproject_-5_data.csv")
```

Check the class of each column

```
#print a summary of the structure of LoanData
str(LoanData)
```

```
## spc_tbl_ [470 × 22] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Variant              : chr [1:470] "Treatment" "Treatment" "Treatment"
"Treatment" ...
##  $ loanofficer_id       : chr [1:470] "qamcqdoe" "qamcqdoe" "qamcqdoe" "qa
mcqdoe" ...
##  $ day                  : num [1:470] 1 2 3 4 5 6 7 8 9 10 ...
##  $ typeI_init           : num [1:470] 0 2 3 1 0 0 0 0 0 0 ...
##  $ typeI_fin            : num [1:470] 0 2 3 2 2 1 1 3 1 0 ...
##  $ typeII_init          : num [1:470] 2 3 0 1 0 4 1 4 4 2 ...
##  $ typeII_fin           : num [1:470] 2 3 0 1 0 0 0 1 1 1 ...
##  $ agree_init           : num [1:470] 7 8 9 8 8 5 8 4 6 9 ...
##  $ agree_fin            : num [1:470] 10 8 9 9 10 10 10 10 10 10 ...
##  $ conflict_init        : num [1:470] 2 2 1 2 2 5 2 6 4 1 ...
##  $ conflict_fin         : num [1:470] 0 2 1 1 0 0 0 0 0 0 ...
##  $ revised_per_ai       : num [1:470] 2 0 0 1 2 5 2 6 4 1 ...
##  $ revised_agst_ai      : num [1:470] 0 0 0 0 0 0 0 0 0 0 ...
##  $ fully_complt         : num [1:470] 9 10 10 10 10 10 10 10 10 10 ...
##  $ confidence_init_total: num [1:470] 706 911 710 694 683 743 993 1000 100
0 1000 ...
##  $ confidence_fin_total : num [1:470] 913 974 970 961 1000 1000 1000 1000
1000 1000 ...
##  $ complt_init          : num [1:470] 9 10 10 10 10 10 10 10 10 10 ...
##  $ complt_fin           : num [1:470] 10 10 10 10 10 10 10 10 10 10 ...
##  $ ai_typeI             : num [1:470] 0 1 2 1 2 1 1 3 1 0 ...
##  $ ai_typeII            : num [1:470] 2 2 0 1 0 0 0 1 1 1 ...
```

```
##  $ badloans_num         : num [1:470] 4 5 2 3 0 4 1 4 4 3 ...
##  $ goodloans_num        : num [1:470] 6 5 8 7 10 6 9 6 6 7 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Variant = col_character(),
##   ..   loanofficer_id = col_character(),
##   ..   day = col_double(),
##   ..   typeI_init = col_double(),
##   ..   typeI_fin = col_double(),
##   ..   typeII_init = col_double(),
##   ..   typeII_fin = col_double(),
##   ..   agree_init = col_double(),
##   ..   agree_fin = col_double(),
##   ..   conflict_init = col_double(),
##   ..   conflict_fin = col_double(),
##   ..   revised_per_ai = col_double(),
##   ..   revised_agst_ai = col_double(),
##   ..   fully_complt = col_double(),
##   ..   confidence_init_total = col_double(),
##   ..   confidence_fin_total = col_double(),
##   ..   complt_init = col_double(),
##   ..   complt_fin = col_double(),
##   ..   ai_typeI = col_double(),
##   ..   ai_typeII = col_double(),
##   ..   badloans_num = col_double(),
##   ..   goodloans_num = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Outliers

Use summary function to see the potential Outliers

```
summary(LoanData)
```

```
##    Variant          loanofficer_id          day         typeI_init
##  Length:470         Length:470         Min.   : 1.0   Min.   : 0.000
##  Class :character   Class :character   1st Qu.: 3.0   1st Qu.: 1.000
##  Mode  :character   Mode  :character   Median : 5.5   Median : 2.000
##                                        Mean   : 5.5   Mean   : 2.619
##                                        3rd Qu.: 8.0   3rd Qu.: 4.000
##                                        Max.   :10.0   Max.   :10.000
##    typeI_fin      typeII_init      typeII_fin       agree_init
##  Min.   :0.000   Min.   :0.000   Min.   :0.0000   Min.   : 0.00
##  1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.: 2.25
##  Median :2.000   Median :1.000   Median :0.0000   Median : 7.00
##  Mean   :1.904   Mean   :1.136   Mean   :0.7298   Mean   : 5.64
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.: 8.00
```

```
##  Max.   :8.000   Max.   :5.000   Max.   :3.0000   Max.   :10.00
##    agree_fin      conflict_init    conflict_fin   revised_per_ai
##  Min.   : 0.000   Min.   :0.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.: 5.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.0000
##  Median : 8.000   Median :2.000   Median :1.000   Median :0.0000
##  Mean   : 6.602   Mean   :1.938   Mean   :1.253   Mean   :0.8149
##  3rd Qu.: 9.000   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :10.000   Max.   :8.000   Max.   :8.000   Max.   :8.0000
##  revised_agst_ai    fully_complt    confidence_init_total confidence_fin_t
otal
##  Min.   :0.00000   Min.   : 0.000   Min.   :  50.0        Min.   :   0.0

##  1st Qu.:0.00000   1st Qu.: 6.000   1st Qu.: 485.5        1st Qu.: 297.2

##  Median :0.00000   Median :10.000   Median : 654.0        Median : 649.5

##  Mean   :0.08511   Mean   : 7.579   Mean   : 624.7        Mean   : 561.9

##  3rd Qu.:0.00000   3rd Qu.:10.000   3rd Qu.: 770.5        3rd Qu.: 810.8

##  Max.   :4.00000   Max.   :10.000   Max.   :1000.0        Max.   :1000.0

##   complt_init      complt_fin        ai_typeI       ai_typeII      badloan
s_num
##  Min.   : 1.00   Min.   : 0.000   Min.   :0.000   Min.   :0.000   Min.   :
0
##  1st Qu.:10.00   1st Qu.: 9.000   1st Qu.:1.000   1st Qu.:0.000   1st Qu.:
2
##  Median :10.00   Median :10.000   Median :2.000   Median :1.000   Median :
3
##  Mean   : 9.47   Mean   : 7.855   Mean   :1.685   Mean   :1.123   Mean   :
3
##  3rd Qu.:10.00   3rd Qu.:10.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:
4
##  Max.   :10.00   Max.   :10.000   Max.   :5.000   Max.   :3.000   Max.   :
5
##  goodloans_num
##  Min.   : 5
##  1st Qu.: 6
##  Median : 7
##  Mean   : 7
##  3rd Qu.: 8
##  Max.   :10
```

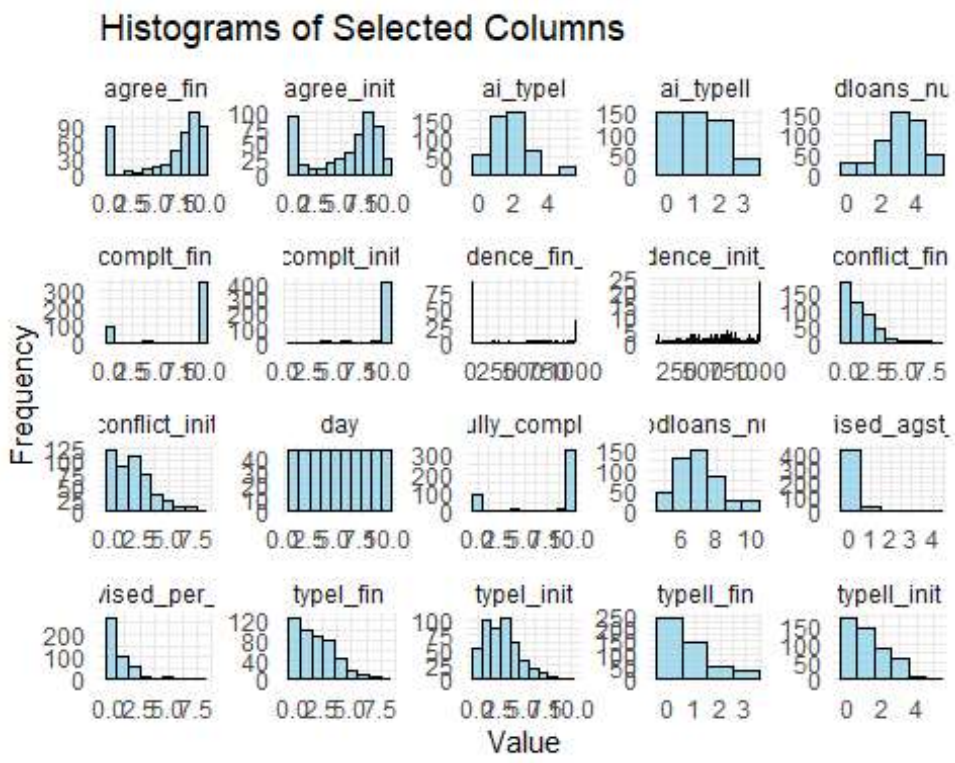
Plot histogram to see potential outliers

```
library(tidyr)
```

```r
# List of columns to plot
columns_to_plot <- c("day", "typeI_init", "typeI_fin", "typeII_init", "typeII
_fin",
                     "agree_init", "agree_fin", "conflict_init", "conflict_fi
n",
                     "revised_per_ai", "revised_agst_ai", "fully_complt",
                     "confidence_init_total", "confidence_fin_total",
                     "complt_init", "complt_fin", "ai_typeI", "ai_typeII",
                     "badloans_num", "goodloans_num")

# Convert data to long format
LoanData_long <- LoanData %>%
  select(all_of(columns_to_plot)) %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Value
")

# Plot histograms using facet_wrap
ggplot(LoanData_long, aes(x = Value)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "black", alpha = 0.
7) +
  facet_wrap(~ Variable, scales = "free") +  # Free scales for better visuali
zation
  labs(title = "Histograms of Selected Columns", x = "Value", y = "Frequency
") +
  theme_minimal()
```



N/A values

Count total number of N/A values

```
na_counts <- colSums(is.na(LoanData))
print(na_counts)
```

```
##               Variant        loanofficer_id                   day
##                     0                     0                     0
##           typeI_init            typeI_fin          typeII_init
##                     0                     0                     0
##           typeII_fin           agree_init            agree_fin
##                     0                     0                     0
##        conflict_init         conflict_fin       revised_per_ai
##                     0                     0                     0
##      revised_agst_ai         fully_complt confidence_init_total
##                     0                     0                     0
##  confidence_fin_total          complt_init            complt_fin
##                     0                     0                     0
##              ai_typeI             ai_typeII          badloans_num
##                     0                     0                     0
##         goodloans_num
##                     0
```

Check duplicate values

```r
duplicates <- LoanData[duplicated(LoanData), ]
print(duplicates)
```

```
## # A tibble: 0 × 22
## # ℹ 22 variables: Variant <chr>, loanofficer_id <chr>, day <dbl>,
## #   typeI_init <dbl>, typeI_fin <dbl>, typeII_init <dbl>, typeII_fin <dbl
>,
## #   agree_init <dbl>, agree_fin <dbl>, conflict_init <dbl>, conflict_fin <
dbl>,
## #   revised_per_ai <dbl>, revised_agst_ai <dbl>, fully_complt <dbl>,
## #   confidence_init_total <dbl>, confidence_fin_total <dbl>, complt_init <
dbl>,
## #   complt_fin <dbl>, ai_typeI <dbl>, ai_typeII <dbl>, badloans_num <dbl>,
## #   goodloans_num <dbl>
```

Check unique value of Variant

```r
unique_values_Variant <- sort(unique(LoanData$Variant))
print(unique_values_Variant)
```

```
## [1] "Control"   "Treatment"
```

Data Exploration

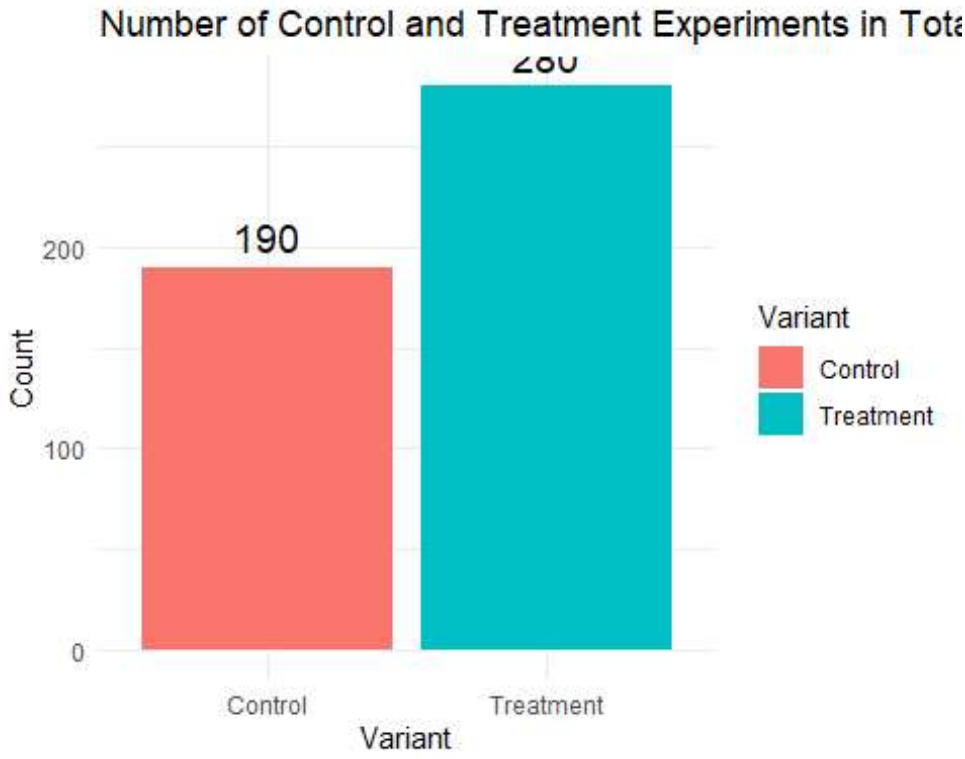
Count Control and Treatment

```r
# Group by Variant and count rows
grouped_counts <- LoanData %>%
  group_by(Variant) %>%
  summarise(Count = n())

# Print the result
print(grouped_counts)
```

```
## # A tibble: 2 × 2
##   Variant   Count
##   <chr>     <int>
## 1 Control     190
## 2 Treatment   280
```

```r
# Plot the grouped counts with labels
ggplot(grouped_counts, aes(x = Variant, y = Count, fill = Variant)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = Count), vjust = -0.5, size = 5) +  # Add labels above
 bars
  labs(title = "Number of Control and Treatment Experiments in Total",
       x = "Variant",
       y = "Count") +
  theme_minimal()
```



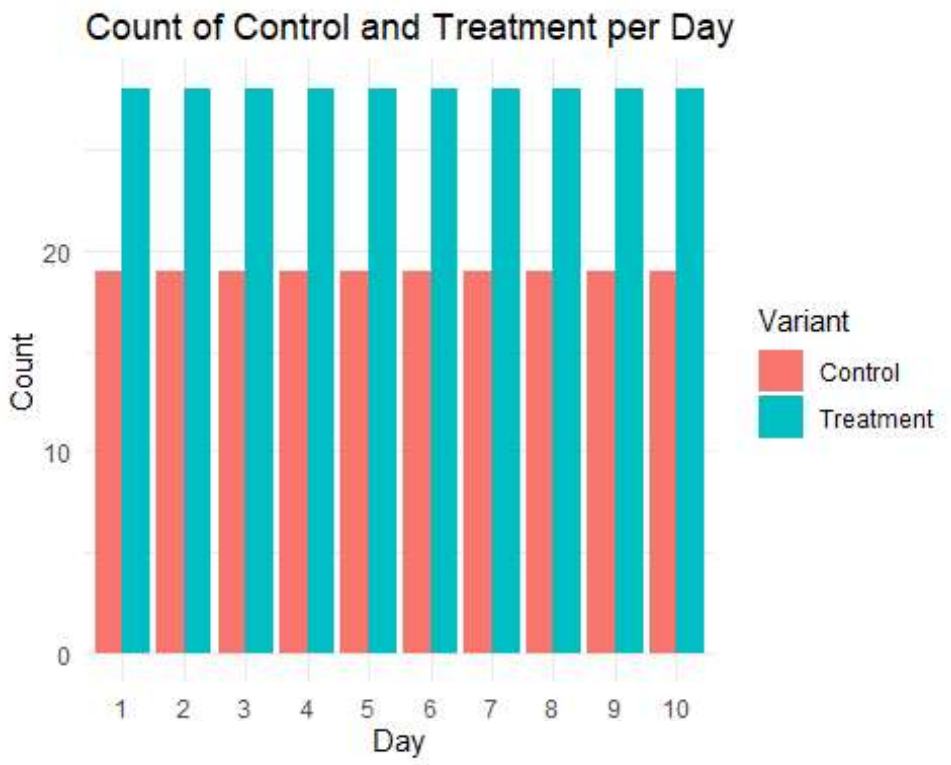

```
# Count the number of each variant (control/treatment) per day
variant_per_day <- LoanData %>%
  group_by(day, Variant) %>%
  summarise(count = n(), .groups = "drop")

# Print the result
print(variant_per_day)
```

```
## # A tibble: 20 × 3
##      day Variant   count
##    <dbl> <chr>     <int>
##  1     1 Control      19
##  2     1 Treatment    28
##  3     2 Control      19
##  4     2 Treatment    28
##  5     3 Control      19
##  6     3 Treatment    28
##  7     4 Control      19
##  8     4 Treatment    28
##  9     5 Control      19
## 10     5 Treatment    28
## 11     6 Control      19
## 12     6 Treatment    28
## 13     7 Control      19
## 14     7 Treatment    28
```

```
## 15     8 Control      19
## 16     8 Treatment    28
## 17     9 Control      19
## 18     9 Treatment    28
## 19    10 Control      19
## 20    10 Treatment    28
```

```r
# Create a bar plot
ggplot(variant_per_day, aes(x = factor(day), y = count, fill = Variant)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Count of Control and Treatment per Day",
       x = "Day",
       y = "Count",
       fill = "Variant") +
  theme_minimal()
```



Check if any loan officers have conducted both control and treatment experiments

```r
# Check if each loan officer has conducted both control and treatment experim
ents
loan_officer_experiment <- LoanData %>%
  group_by(loanofficer_id) %>%
  summarise(experiment_types = n_distinct(Variant)) %>%
  mutate(Control_and_Treatment = ifelse(experiment_types > 1, "Both Control a
nd Treatment", "Only One experiment"))
```

```r
# Print the result
print(loan_officer_experiment)
```

```
## # A tibble: 47 × 3
##    loanofficer_id experiment_types Control_and_Treatment
##    <chr>                     <int> <chr>
##  1 0899qxvc                      1 Only One experiment
##  2 09pij0e2                      1 Only One experiment
##  3 0g7pi6g8                      1 Only One experiment
##  4 0gh7r2hr                      1 Only One experiment
##  5 1ckkyukp                      1 Only One experiment
##  6 1ha5khxo                      1 Only One experiment
##  7 2twvlktb                      1 Only One experiment
##  8 2udootyt                      1 Only One experiment
##  9 4cdwcblq                      1 Only One experiment
## 10 530lfgx0                      1 Only One experiment
## # ℹ 37 more rows
```

Check total of control or treatment experiments per 1 officer

```r
# Count the number of Control and Treatment experiments per loan officer
experiment_count_per_officer <- LoanData %>%
  group_by(loanofficer_id, Variant) %>%
  summarise(experiment_count = n(), .groups = "drop")

# Print the result
print(experiment_count_per_officer)
```

```
## # A tibble: 47 × 3
##    loanofficer_id Variant   experiment_count
##    <chr>          <chr>                <int>
##  1 0899qxvc       Treatment               10
##  2 09pij0e2       Treatment               10
##  3 0g7pi6g8       Control                 10
##  4 0gh7r2hr       Control                 10
##  5 1ckkyukp       Treatment               10
##  6 1ha5khxo       Treatment               10
##  7 2twvlktb       Treatment               10
##  8 2udootyt       Control                 10
##  9 4cdwcblq       Treatment               10
## 10 530lfgx0       Treatment               10
## # ℹ 37 more rows
```

Check unique value of loanofficer_id

```r
unique_count_loanofficer_id <- length(unique(LoanData$loanofficer_id))
print(unique_count_loanofficer_id)
```

```
## [1] 47
```

Check the total value of badloan and goodloan

```
LoanData_grouped <- LoanData %>%
  mutate(total_loans = badloans_num + goodloans_num) %>%
  group_by(total_loans) %>%
  summarise(count = n())

# Print the result
print(LoanData_grouped)
```

```
## # A tibble: 1 × 2
##   total_loans count
##         <dbl> <int>
## 1          10   470
```

```
LoanData %>%
  group_by(Variant) %>%
  summarise(avg_good = mean(goodloans_num), avg_bad = mean(badloans_num))
```

```
## # A tibble: 2 × 3
##   Variant   avg_good avg_bad
##   <chr>        <dbl>   <dbl>
## 1 Control          7       3
## 2 Treatment        7       3
```

Check the total loans that loan officer has made the decision per day in the final stage (complt_fin)

```
filtered_complt_fin10 <- LoanData %>%
  filter(complt_fin != 10) %>%
  group_by(Variant) %>%
  summarise(count = n())

# Print the result
print(filtered_complt_fin10)
```

```
## # A tibble: 2 × 2
##   Variant   count
##   <chr>     <int>
## 1 Control      97
## 2 Treatment    23
```

Check the total value of “agree_fin”, “conflict_fin”

```
# Sum of agree_fin and conflict_fin
LoanData <- LoanData %>%
  mutate(total_AgreeConflict = agree_fin+conflict_fin)
```

```r
# Check if any rows have total_loans not equal to 10
rows_AgreeConflict <- LoanData %>%
  filter(total_AgreeConflict ==0)

# Print the rows where the sum of "agree_fin", "conflict_fin" is 0
print(rows_AgreeConflict)
```

```
## # A tibble: 90 × 23
##    Variant loanofficer_id   day typeI_init typeI_fin typeII_init typeII_fi
n
##    <chr>   <chr>          <dbl>      <dbl>     <dbl>       <dbl>      <dbl
>
##  1 Control 2udootyt           1          2         0           1
0
##  2 Control 2udootyt           2          4         0           1
0
##  3 Control 2udootyt           3          3         0           2
0
##  4 Control 2udootyt           4          3         0           2
0
##  5 Control 2udootyt           5          3         0           0
0
##  6 Control 2udootyt           6          5         0           0
0
##  7 Control 2udootyt           7          5         0           0
0
##  8 Control 2udootyt           8          4         0           1
0
##  9 Control 2udootyt           9          4         0           0
0
## 10 Control 2udootyt          10          2         0           0
0
## # ℹ 80 more rows
## # ℹ 16 more variables: agree_init <dbl>, agree_fin <dbl>, conflict_init <d
bl>,
## #   conflict_fin <dbl>, revised_per_ai <dbl>, revised_agst_ai <dbl>,
## #   fully_complt <dbl>, confidence_init_total <dbl>,
## #   confidence_fin_total <dbl>, complt_init <dbl>, complt_fin <dbl>,
## #   ai_typeI <dbl>, ai_typeII <dbl>, badloans_num <dbl>, goodloans_num <db
l>,
## #   total_AgreeConflict <dbl>
```

Filter rows that complete the decision without using AI at all

```r
# Filter rows based on conditions
filtered_rows <- LoanData %>%
```

```
  filter(complt_init > 0 & complt_fin == 0)

# View the filtered rows
print(filtered_rows)
```

```
## # A tibble: 90 × 23
##    Variant loanofficer_id   day typeI_init typeI_fin typeII_init typeII_fi
n
##    <chr>   <chr>          <dbl>      <dbl>     <dbl>       <dbl>      <dbl
>
##  1 Control 2udootyt           1          2         0           1
0
##  2 Control 2udootyt           2          4         0           1
0
##  3 Control 2udootyt           3          3         0           2
0
##  4 Control 2udootyt           4          3         0           2
0
##  5 Control 2udootyt           5          3         0           0
0
##  6 Control 2udootyt           6          5         0           0
0
##  7 Control 2udootyt           7          5         0           0
0
##  8 Control 2udootyt           8          4         0           1
0
##  9 Control 2udootyt           9          4         0           0
0
## 10 Control 2udootyt          10          2         0           0
0
## # ℹ 80 more rows
## # ℹ 16 more variables: agree_init <dbl>, agree_fin <dbl>, conflict_init <d
bl>,
## #   conflict_fin <dbl>, revised_per_ai <dbl>, revised_agst_ai <dbl>,
## #   fully_complt <dbl>, confidence_init_total <dbl>,
## #   confidence_fin_total <dbl>, complt_init <dbl>, complt_fin <dbl>,
## #   ai_typeI <dbl>, ai_typeII <dbl>, badloans_num <dbl>, goodloans_num <db
l>,
## #   total_AgreeConflict <dbl>
```

Confidence score

```
### Confidence score

LoanData %>%
  select(complt_fin,confidence_fin_total)
```

```
## # A tibble: 470 × 2
##    complt_fin confidence_fin_total
##         <dbl>                <dbl>
##  1         10                  913
##  2         10                  974
##  3         10                  970
##  4         10                  961
##  5         10                 1000
##  6         10                 1000
##  7         10                 1000
##  8         10                 1000
##  9         10                 1000
## 10         10                 1000
## # ℹ 460 more rows
```

####Sum total value of confidence score in the final stage from all rows where sum of "agree_fin", "conflict_fin" is 0

```
total_confidence_0 <- rows_AgreeConflict %>%
  summarise(total = sum(confidence_fin_total, na.rm = TRUE))

print(total_confidence_0)
```

```
## # A tibble: 1 × 1
##   total
##   <dbl>
## 1     0
```

####Sum total value of confidence score in the final stage from all rows where complt_init > 0 & complt_fin == 0

```
total_confidence_withoutAI <- filtered_rows %>%
  summarise(total = sum(confidence_fin_total, na.rm = TRUE))

print(total_confidence_withoutAI)
```

```
## # A tibble: 1 × 1
##   total
##   <dbl>
## 1     0
```

Filter rows that the sum of agree and conflict do not add up to complt_fin AND involves AI in the decision

```
filtered_agr_conf <- LoanData %>%
  filter((agree_init + conflict_init != complt_init) | (agree_fin + conflict_
fin != complt_fin))
```

```
filtered_agr_conf_AI <- filtered_agr_conf %>%
  filter(!(complt_init > 0 & complt_fin == 0))

# View the filtered rows
#print(filtered_agr_conf_AI )

filtered_agr_conf_AI %>%
  select(Variant, loanofficer_id, agree_init, conflict_init, complt_init, agr
ee_fin,conflict_fin,complt_fin) %>%
  print()
```

```
## # A tibble: 6 × 8
##   Variant   loanofficer_id agree_init conflict_init complt_init agree_fin
##   <chr>     <chr>                <dbl>         <dbl>       <dbl>     <dbl>
## 1 Treatment 92vdohom                 9             0          10         7
## 2 Control   qwun9ha5                 4             4          10         6
## 3 Treatment envu2p1p                 6             1           8         6
## 4 Treatment envu2p1p                 3             3          10         6
## 5 Treatment envu2p1p                 5             3          10         6
## 6 Treatment 9lejzokf                 1             7          10         6
## # ℹ 2 more variables: conflict_fin <dbl>, complt_fin <dbl>
```

Filter out rows that complete the decision without using AI at all, as the A/B testing is focusing on whether the new computer model actually improves the loan officers' decision quality or not. Thus, the decision without AI will be excluded.

```
# Remove rows where complt_init > 0 and complt_fin == 0
df <- LoanData %>%
  filter(!(complt_init > 0 & complt_fin == 0))

# View the remaining rows
print(df)
```

```
## # A tibble: 380 × 23
##    Variant   loanofficer_id   day typeI_init typeI_fin typeII_init typeII_
fin
##    <chr>     <chr>          <dbl>      <dbl>     <dbl>       <dbl>      <d
bl>
##  1 Treatment qamcqdoe           1          0         0           2
  2
##  2 Treatment qamcqdoe           2          2         2           3
  3
##  3 Treatment qamcqdoe           3          3         3           0
  0
##  4 Treatment qamcqdoe           4          1         2           1
  1
##  5 Treatment qamcqdoe           5          0         2           0
  0
```

```
##  6 Treatment qamcqdoe           6          0         1           4
  0
##  7 Treatment qamcqdoe           7          0         1           1
  0
##  8 Treatment qamcqdoe           8          0         3           4
  1
##  9 Treatment qamcqdoe           9          0         1           4
  1
## 10 Treatment qamcqdoe          10          0         0           2
  1
## # ℹ 370 more rows
## # ℹ 16 more variables: agree_init <dbl>, agree_fin <dbl>, conflict_init <d
bl>,
## #   conflict_fin <dbl>, revised_per_ai <dbl>, revised_agst_ai <dbl>,
## #   fully_complt <dbl>, confidence_init_total <dbl>,
## #   confidence_fin_total <dbl>, complt_init <dbl>, complt_fin <dbl>,
## #   ai_typeI <dbl>, ai_typeII <dbl>, badloans_num <dbl>, goodloans_num <db
l>,
## #   total_AgreeConflict <dbl>
```

```
control_count <- sum(df$Variant == "Control")
treatment_count <- sum(df$Variant == "Treatment")

print(control_count)
```

```
## [1] 100
```

```
print(treatment_count)
```

```
## [1] 280
```

Appendix B: Data Analysis: Hypothesis Testing

```
## Set categorical variable
df$Variant <- factor(df$Variant)
```

OEC no.1 : Type I and Type II Error rate

First, we can evaluate the performance of the new model using the following metrics as OEC (Overall Evaluation Criteria):

1) Type I Error Rate = False Positives / Total Actual Negatives = typeI_fin / goodloans_num
- Hypothesis: The new model will reduce the Type I Error Rate – single tailed t-test
2) Type II Error Rate = False Negatives / Total Actual Positives = typeII_fin / badloans_num
- Hypothesis: The new model will reduce the Type II Error Rate – single tailed t-test

Both errors are considered after the officer reviews the computer prediction (as it is in the current process).

```
df1<-df

# Calculate Type I and Type II error rates aggregating the information by loa
n officer ID

df1 <- df1 %>% group_by(Variant,loanofficer_id) %>%
  summarise(
    Type_I_Error_Rate = sum(typeI_fin) / sum(goodloans_num),  # False Positiv
es / Total Actual Negatives


    Type_II_Error_Rate = sum(typeII_fin) / sum(badloans_num)  # False Negativ
es / Total Actual Positives
  )

## `summarise()` has grouped output by 'Variant'. You can override using the
## `.groups` argument.

df1 <- df1 %>%
  mutate(
    Type_I_Error_Rate = ifelse(is.na(Type_I_Error_Rate),0,Type_I_Error_Rate),
  # NA with 0
    Type_II_Error_Rate = ifelse(is.na(Type_II_Error_Rate),0,Type_II_Error_Rat
e)  # NA with 0
  )
```

Run Welch's two-sample t-tests to examine if there's sig. difference between pairs of Variants

Type_I_Error_Rate

```
t.test(
  Type_I_Error_Rate ~ Variant,
  data = df1,
  var.equal = FALSE) # assuming samples have unequal variances (using Welch t
-test)

##
##  Welch Two Sample t-test
##
## data:  Type_I_Error_Rate by Variant
## t = 4.0648, df = 11.019, p-value = 0.001861
## alternative hypothesis: true difference in means between group Control and
 group Treatment is not equal to 0
## 95 percent confidence interval:
##  0.1035696 0.3480630
## sample estimates:
##   mean in group Control mean in group Treatment
##               0.5028571               0.2770408
```

Type_II_Error_Rate

```
t.test(
  Type_II_Error_Rate ~ Variant,
  data = df1,
  var.equal = FALSE) # assuming samples have unequal variances (using Welch t
-test)

##
##  Welch Two Sample t-test
##
## data:  Type_II_Error_Rate by Variant
## t = 3.4409, df = 10.923, p-value = 0.005571
## alternative hypothesis: true difference in means between group Control and
 group Treatment is not equal to 0
## 95 percent confidence interval:
##  0.04514467 0.20580771
## sample estimates:
##   mean in group Control mean in group Treatment
##               0.3933333               0.2678571
```

Compute the difference (in percentage) with each OEC between Variants

```
# Compute mean OEC for each Variant
mean_OEC_each_Variant <- df1 %>%
```

```
  group_by(Variant) %>%
  summarise(mean_Type_I_Error_Rate = mean(Type_I_Error_Rate),
            mean_Type_II_Error_Rate = mean(Type_II_Error_Rate))

# View mean OEC
print(mean_OEC_each_Variant)
```

```
## # A tibble: 2 × 3
##   Variant   mean_Type_I_Error_Rate mean_Type_II_Error_Rate
##   <fct>                      <dbl>                   <dbl>
## 1 Control                    0.503                   0.393
## 2 Treatment                  0.277                   0.268
```

```
# Compute pairwise % differences in OEC between pairs of variants
pairwise_diff <- mean_OEC_each_Variant %>%
  summarise(
    Dif_Type_I_Error_Rate = mean_Type_I_Error_Rate[Variant == "Treatment"] -
mean_Type_I_Error_Rate[Variant == "Control"],
    Dif_Type_II_Error_Rate = mean_Type_II_Error_Rate[Variant == "Treatment"]
- mean_Type_II_Error_Rate[Variant == "Control"],
    Perc_Type_I_Error_Rate = (Dif_Type_I_Error_Rate / mean_Type_I_Error_Rate
[Variant == "Control"]) * 100,
    Perc_Type_II_Error_Rate = (Dif_Type_II_Error_Rate / mean_Type_II_Error_Ra
te[Variant == "Control"]) * 100
  )

perc_dif <- pairwise_diff %>% select(Perc_Type_I_Error_Rate,Perc_Type_II_Erro
r_Rate)

# View pairwise differences
print(perc_dif)
```

```
## # A tibble: 1 × 2
##   Perc_Type_I_Error_Rate Perc_Type_II_Error_Rate
##                    <dbl>                   <dbl>
## 1                  -44.9                   -31.9
```

Compute & Interpret Effect Size (Cohen's d)

Effect size: Control vs Treatment

Type_I_Error_Rate

```
Control = df1$Type_I_Error_Rate[df1$Variant == "Control"]
Treatment = df1$Type_I_Error_Rate[df1$Variant == "Treatment"]

cohens_d(Treatment, Control) # compute effect size of difference between Trea
tment & Control
```

```
## Cohen's d |         95% CI
## -------------------------
## -1.96     | [-2.80, -1.10]
##
## - Estimated using pooled SD.

effectsize::interpret_cohens_d(-1.96)

## [1] "large"
## (Rules: cohen1988)
```

Type_II_Error_Rate

```
Control = df1$Type_II_Error_Rate[df1$Variant == "Control"]
Treatment = df1$Type_II_Error_Rate[df1$Variant == "Treatment"]

cohens_d(Treatment, Control) # compute effect size of difference between Trea
tment & Control

## Cohen's d |         95% CI
## -------------------------
## -1.67     | [-2.48, -0.84]
##
## - Estimated using pooled SD.

effectsize::interpret_cohens_d(-1.67)

## [1] "large"
## (Rules: cohen1988)
```

Metric no.2 : Ratio of Loan officer's agreements with computer predictions after seeing computer predictions

```
# Aggregate data by loanofficer_id and Variant
df2 <- df %>%
  group_by(loanofficer_id, Variant) %>%
  summarise(
    agree_fin = sum(agree_fin, na.rm = TRUE),
    complt_fin = sum(complt_fin, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(agree_ratio = agree_fin / complt_fin)
```

Run Welch's two-sample t-tests to examine if there's sig. difference between pairs of Variants

```
# Independent t-test
t_test_agreement_ratio <- t.test(agree_ratio ~ Variant, data = df2, var.equal
```

```
 = FALSE)
print(t_test_agreement_ratio)

##
##  Welch Two Sample t-test
##
## data:  agree_ratio by Variant
## t = -3.4329, df = 11.383, p-value = 0.005333
## alternative hypothesis: true difference in means between group Control and
 group Treatment is not equal to 0
## 95 percent confidence interval:
##  -0.20213704 -0.04459309
## sample estimates:
##   mean in group Control mean in group Treatment
##               0.7491300               0.8724951
```

Compute & Interpret Effect Size (Cohen's d)

```
# Effect size (Cohen's d)
Control_Agreement_Ratio <- df2$agree_ratio[df2$Variant == "Control"]
Treatment_Agreement_Ratio <- df2$agree_ratio[df2$Variant == "Treatment"]
cohens_d(Control_Agreement_Ratio , Treatment_Agreement_Ratio)

## Cohen's d |         95% CI
## -------------------------
## -1.60     | [-2.40, -0.78]
##
## - Estimated using pooled SD.

effectsize::interpret_cohens_d(-1.60)

## [1] "large"
## (Rules: cohen1988)
```

Metric no.3 : Loan Officer Confidence

Loan officer confidence

```
# Hypothesis: Higher confidence_fin_total in Treatment suggests improved mode
l effectiveness
```

Run Welch's two-sample t-tests to examine if there's sig. difference between pairs of Variants

```
# Independent t-test
t_test_confidence <- t.test(confidence_fin_total ~ Variant, data = df, var.eq
ual = FALSE)
print(t_test_confidence)
```

```
##
##  Welch Two Sample t-test
##
## data:  confidence_fin_total by Variant
## t = -6.0989, df = 222.19, p-value = 4.687e-09
## alternative hypothesis: true difference in means between group Control and
 group Treatment is not equal to 0
## 95 percent confidence interval:
##  -179.29481  -91.72233
## sample estimates:
##   mean in group Control mean in group Treatment
##                595.1200                730.6286
```

Compute & Interpret Effect Size (Cohen's d)

```
# Effect size (Cohen's d)
Control_Confidence <- df$confidence_fin_total[df$Variant == "Control"]
Treatment_Confidence <- df$confidence_fin_total[df$Variant == "Treatment"]
cohens_d(Control_Confidence ,Treatment_Confidence)

## Cohen's d |         95% CI
## -------------------------
## -0.63     | [-0.86, -0.40]
##
## - Estimated using pooled SD.

effectsize::interpret_cohens_d(-0.63)

## [1] "medium"
## (Rules: cohen1988)
```

Metric no.4 : Revised per AI

```
# Aggregate data at the loan officer level

df4 <- df %>%
  group_by(loanofficer_id, Variant) %>%
  summarise(revised_ratio = sum(revised_per_ai) / sum(complt_fin), .groups =
"drop")
```

Run Welch's two-sample t-tests to examine if there's sig. difference between pairs of Variants

```
# Perform t-test on the aggregated data
t_test_revised_ratio <- t.test(
  revised_ratio ~ Variant,
  data = df4,
  var.equal = FALSE
)
```

```
print(t_test_revised_ratio)
```

```
##
##  Welch Two Sample t-test
##
## data:  revised_ratio by Variant
## t = -2.9706, df = 34.614, p-value = 0.005371
## alternative hypothesis: true difference in means between group Control and
 group Treatment is not equal to 0
## 95 percent confidence interval:
##  -0.10901678 -0.02048277
## sample estimates:
##   mean in group Control mean in group Treatment
##              0.05545891              0.12020869
```

Compute & Interpret Effect Size (Cohen's d)

```
# Compute Cohen's d for revised_ratio
Control_Revised_Ratio <- df4$revised_ratio[df4$Variant == "Control"]
Treatment_Revised_Ratio <- df4$revised_ratio[df4$Variant == "Treatment"]
cohens_d(Control_Revised_Ratio,Treatment_Revised_Ratio)
```

```
## Cohen's d |         95% CI
## -------------------------
## -0.77     | [-1.51, -0.02]
##
## - Estimated using pooled SD.
```

```
effectsize::interpret_cohens_d(-0.77)
```

```
## [1] "medium"
## (Rules: cohen1988)
```

UNFILTERED DATA

OEC no.1 : Type II Error rate

Type II Error Rate = False Negatives / Total Actual Positives = typeII_fin / badloans_num - Hypothesis: The new model will reduce the Type II Error Rate – single tailed t-test

```
df5<-LoanData

# Calculate Type I and Type II error rates aggregating the information by loa
n officer ID

df5 <- df5 %>% group_by(Variant,loanofficer_id) %>%
  summarise(
    Type_II_Error_Rate = sum(typeII_fin) / sum(badloans_num)  # False Negativ
```

```
es / Total Actual Positives
  )

## `summarise()` has grouped output by 'Variant'. You can override using the
## `.groups` argument.

df5 <- df5 %>%
  mutate(
    Type_II_Error_Rate = ifelse(is.na(Type_II_Error_Rate),0,Type_II_Error_Rat
e)  # NA with 0
  )
```

Run Welch's two-sample t-tests to examine if there's sig. difference between pairs of Variants

Type_II_Error_Rate

```
t.test(
  Type_II_Error_Rate ~ Variant,
  data = df5,
  var.equal = FALSE) # assuming samples have unequal variances (using Welch t
-test)

##
##  Welch Two Sample t-test
##
## data:  Type_II_Error_Rate by Variant
## t = -1.1967, df = 19.84, p-value = 0.2455
## alternative hypothesis: true difference in means between group Control and
 group Treatment is not equal to 0
## 95 percent confidence interval:
##  -0.16694635  0.04526716
## sample estimates:
##   mean in group Control mean in group Treatment
##               0.2070175               0.2678571
```

Compute & Interpret Effect Size (Cohen's d)

Effect size: Control vs Treatment

Type_II_Error_Rate

```
Control = df5$Type_II_Error_Rate[df5$Variant == "Control"]
Treatment = df5$Type_II_Error_Rate[df5$Variant == "Treatment"]

cohens_d(Treatment, Control) # compute effect size of difference between Trea
tment & Control
```

```
## Cohen's d |        95% CI
## -------------------------
## 0.42      | [-0.17, 1.01]
##
## - Estimated using pooled SD.
```

```
effectsize::interpret_cohens_d(0.42)
```

```
## [1] "small"
## (Rules: cohen1988)
```

Estimate the appropriate sample size

With 2 Variants (2 Samples/Groups)

```
#Estimate sample size
pwr.t.test(power = .8, # 80% power
           d = 0.2, # Cohen's d
           sig.level = 0.05, # threshold for p-val
           type = "two.sample") # eg., this is for treatment vs control
```

```
##
##      Two-sample t test power calculation
##
##               n = 393.4057
##               d = 0.2
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Appendix C: Tables and Figures

Tables

The following tables are referenced in the main report:

- **Table 1:** Type II Error (Refer to Section 4.5.1 for details).
- **Table 2:** Type I Error (Refer to Section 4.5.2 for details).
- **Table 3:** Loan Officer Agreement with Model (Refer to Section 4.5.3 for details).
- **Table 4:** Loan Officer Final Confidence (Refer to Section 4.5.4 for details).
- **Table 5:** Loan Officer Decision Revisions (Refer to Section 4.5.5 for details).

Figures

The following figures are included in the main report:

- **Figure 1:** Number of Control and Treatment Experiments in Total (Refer to Section 4.1 for details).
- **Figure 2:** Count of Control and Treatment Per Day (Refer to Section 4.1 for details).