Data Science for Marketing: A/B Testing and Statistical Analysis

Silverblue Group

April 1, 2025

Course Information and Instructor

- Course: 2025 Spring MAP 2192-001-15022 Mathematics of Data Science
- Instructor: Dr. Zhijun Yin

Group Member List

Name	Contribution
CJ Bingham	Everything

Title of the Project

Data Science for Marketing: A/B Testing and Statistical Analysis

Motivation

"Data-Driven" decision making is crucial for companies to make informed decisions regarding marketing and ad campaigns. As data analysts, we must understand how to find statistically significant relationships between different variables, and create visualizations of those relationships. We can create hypothesis tests to look at different scenarios, and Z-tests can help to show statistical relationships. Logistic regression analysis is a tool used to analyze relationships between different variables.

We take a sample dataset containing 5000 rows of marketing data. Columns include data such as marketing data campaign id, positive or negative response from the user, and the percent grade for each customer's click through rate. We then perform analysis on the dataset to gather insights to better focus our marketing strategies and better understand our user. This can help increase the return on investment of ad spending.

Tools Used

- Definitions and principles of hypothesis testing and the Z test
- Definitions and principles of logistic regression
- Python, Jupyter, and LaTex for implementation
- Python packages for data manipulation, analysis, and visualization

Solving Process

1. Our Hypothesis Statement and A/B Test

An A/B test takes two different campaigns shown to a user and changes a certain variable in one of the campaigns. This could be a different ad or call to action statement. In this case we look at two different sets of campaigns, one from the first half of campaigns, and one from the second half of campaigns. We are conducting a hypothesis test to compare the response rates of two different sets of marketing campaigns.

- Null Hypothesis (H_0): The response rate of the control group is equal to the response rate of the treatment group.

$$H_0 : p_{control} = p_{treatment}$$

- Alternative Hypothesis (H_1): The response rate of the treatment group is higher than the control group.

$$H_1 : p_{control} < p_{treatment}$$

Decision rule: Reject H_0 if p-value $< \alpha$ (typically 0.05)

Z-Test for Comparing Response Rates

To determine whether there is a statistically significant difference in positive response rates between the control and treatment groups, we conduct a two-proportion Z-test. The test statistic is calculated as:

$$z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \tag{1}$$

Where:

- $\hat{p}_1 = \frac{x_1}{n_1}$ is the sample proportion of positive responses in the control group.
- $\hat{p}_2 = \frac{x_2}{n_2}$ is the sample proportion of positive responses in the treatment group.
- n_1, n_2 are the sample sizes of the control and treatment groups, respectively.
- x_1, x_2 are the number of successful responses in each group.
- $\hat{p}$ is the pooled sample proportion, given by:

$$\hat{p} = \frac{x_1 + x_2}{n_1 + n_2} \tag{2}$$

This test helps assess whether the observed difference in conversion rates is statistically significant or could have occurred by chance.

Python Code

```
# Import necessary libraries
import pandas as pd
import numpy as np
from statsmodels.stats.proportion import proportions_ztest
from statsmodels.stats.proportion import proportion_confint
import matplotlib.pyplot as plt
# Create dataframe from dataset and create control_group and
                                  treatment_group
df = pd.read_csv("marketing_dataset.csv")
# Control_group contains campaigns 1-9
control_group = df[(df['campaign_id'].isin(tuple(f'CAMPAIGN_{i}'
                                  for i in range(1, 10))))]
# Treatment_group contains campaigns 11-19
treatment_group = df[(df['campaign_id'].isin(tuple(f'CAMPAIGN_{i}'
                                  for i in range(11, 20))))]
# Find total users in each group
control_total = len(control_group)
treatment_total = len(treatment_group)
print(f"Total number of users in first half of marketing campaigns:
                                   {control_total}")
print(f"Total number of users in in second half of marketing
                                  campaigns: {treatment_total}")
# Compute conversions
control_conversions = (control_group['customer_response'] == 1).sum
                                  ()
treatment_conversions = (treatment_group['customer_response'] == 1)
                                  .sum()
print(f"Total number of positive responses in first half of
                                  marketing campaigns: {
                                  control_conversions}")
print(f"Total number of positive responses in second half of
                                  marketing campaigns: {
                                  treatment_conversions}")
def ab_test_analysis(control_conversions, control_total,
```

```python
                     treatment_conversions, treatment_total):
    # Calculate proportions
    p_control = control_conversions / control_total
    p_treat = treatment_conversions / treatment_total

    # Perform z-test
    count = np.array([control_conversions, treatment_conversions])
    nobs = np.array([control_total, treatment_total])
    z_stat, p_val = proportions_ztest(count, nobs)

    # Calculate confidence intervals
    ci_control = proportion_confint(control_conversions,
                                      control_total, method='normal
                                      ')
    ci_treat = proportion_confint(treatment_conversions,
                                      treatment_total, method='
                                      normal')

    fig, ax = plt.subplots(figsize=(6, 5))
    groups = ['Control Group', 'Treatment Group']
    proportions = [p_control, p_treat]
    ci_lower = [ci_control[0], ci_treat[0]]
    ci_upper = [ci_control[1], ci_treat[1]]

    # Create bar plot with error bars
    ax.bar(groups, proportions, yerr=[
        [p - l for p, l in zip(proportions, ci_lower)],
        [u - p for u, p in zip(ci_upper, proportions)]
    ], capsize=10, color=['blue', 'red'], alpha=0.6)

    # Labels and title
    ax.set_ylabel("Response Rate")
    ax.set_title("Comparison of Response Rates with 95% CIs")
    ax.axhline(y=p_control, color='blue', linestyle='dashed', alpha
                                      =0.7, label="Control Mean")
    ax.axhline(y=p_treat, color='red', linestyle='dashed', alpha=0.
                                      7, label="Treatment Mean")
    ax.legend()
    plt.savefig("CI.png")
    # Show plot
    plt.show()

    return {
        'z_test_statistic': f"{z_stat:.2f}",
        'p_value_z_test': f"{p_val:.2f}",
        'control_response_rate': f"{p_control:.2f}",
        'treatment_response_rate': f"{p_treat:.2f}",
        'control_response_rate_confidence_interval': ci_control,
        'treatment_response_rate_confidence_interval': ci_treat
    }
ab_test_analysis(control_conversions, control_total,
                                  treatment_conversions,
                                  treatment_total)
```

Statistical Test and Results

We perform a two-proportion Z-test to determine if there is a statistically significant difference between the response rates of the two groups. Below are the test results:

Metric	**Control Group**	**Treatment Group**
Sample Size (n)	2213	2266
Successful Responses (x)	409	475
Response Rate (p)	0.18	0.21
95% Confidence Interval	(0.169, 0.201)	(0.193, 0.226)
Z-test Statistic	-2.09	
P-value	0.04	

Table 1: Summary of A/B Test Results

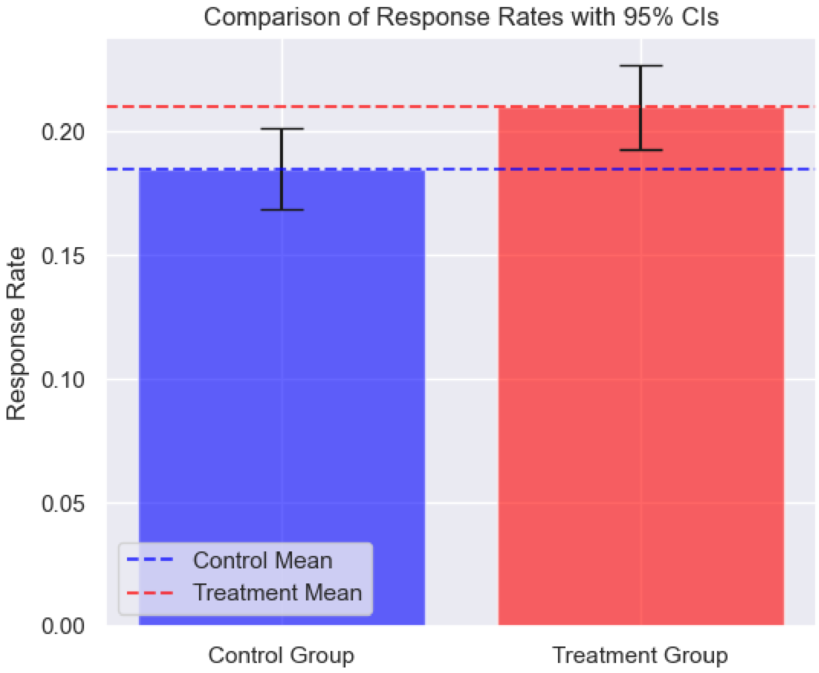


Figure 1: Overlap of 95 percent confidence intervals

Statistical Conclusion

With a p-value of 0.04 (below the significance level of 0.05), we reject the null hypothesis. This suggests that the treatment group has a significantly higher response rate than the control group. The observed increase in response rate indicates a potentially effective marketing strategy.

2. Introduction to Logistic Regression Analysis

Logistic regression analysis is a process used to estimate the probability of an event occurring, such as a conversion or clicking on an ad. With our marketing dataset, we use the analysis to compare two variables, click through rate and customer response. This helps to predict customer behavior and help refine marketing efforts.

Logistic Regression Model

For predicting binary outcomes (positive or negative response):

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k \tag{3}$$

Where:

- p = Probability of positive response
- β_i = Coefficients
- X_i = Predictor variable (click through rate)

The target column in the dataset is customer response, with a value of 1 for a positive response, and a 0 for a negative response. We use the following Python code to create a perform the logistic regression analysis:

Python Code

```python
# Import necessary libraries
import pandas as pd
import numpy as np
import statsmodels.api as sm
import seaborn as sns
import matplotlib.pyplot as plt

# Load data
df = pd.read_csv("marketing_dataset.csv")

# Define predictor (X) and target (y)
X = df["click_through_rate"]
y = df["customer_response"]
```

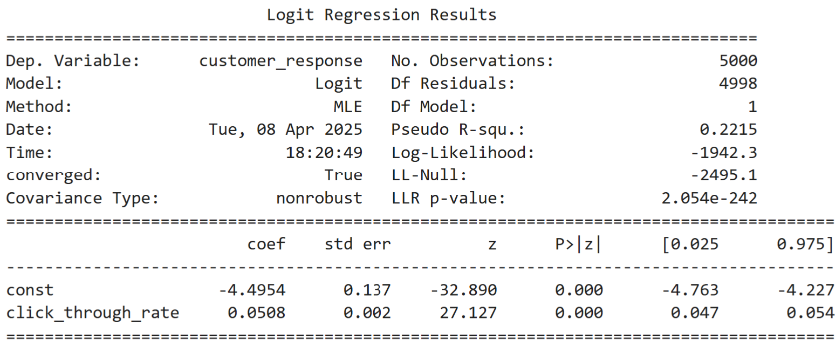
```python
# Fit logistic regression model
X_const = sm.add_constant(X)  # Add intercept
logit_model = sm.Logit(y, X_const).fit()
print(logit_model.summary())

# Generate prediction data
X_range = np.linspace(X.min(), X.max(), 300)
X_range_const = sm.add_constant(X_range)  # Add intercept for prediction
y_pred = logit_model.predict(X_range_const)

# Plot scatterplot of actual data
plt.figure(figsize=(8, 6))
sns.scatterplot(x=X, y=y, alpha=0.3, color="blue", label="Data")

# Plot logistic regression curve
plt.plot(X_range, y_pred, color="red", linewidth=2, label="Logistic Regression Curve")

# Labels and title
plt.xlabel("Click Through Rate Rating for Customer (%)")
plt.ylabel("Customer Response (0 = No, 1 = Yes)")
plt.title("Logistic Regression Test: Click Through Rate vs Customer Response")
plt.legend()
plt.savefig("log_regression.png")
plt.show()
```



```
                           Logit Regression Results
==============================================================================
Dep. Variable:      customer_response   No. Observations:                 5000
Model:                          Logit   Df Residuals:                     4998
Method:                           MLE   Df Model:                            1
Date:                Tue, 08 Apr 2025   Pseudo R-squ.:                  0.2215
Time:                        18:20:49   Log-Likelihood:                -1942.3
converged:                       True   LL-Null:                       -2495.1
Covariance Type:            nonrobust   LLR p-value:                2.054e-242
======================================================================================
                         coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------
const                 -4.4954      0.137    -32.890      0.000      -4.763      -4.227
click_through_rate     0.0508      0.002     27.127      0.000       0.047       0.054
======================================================================================
```

Figure 2: Logistic Regression results

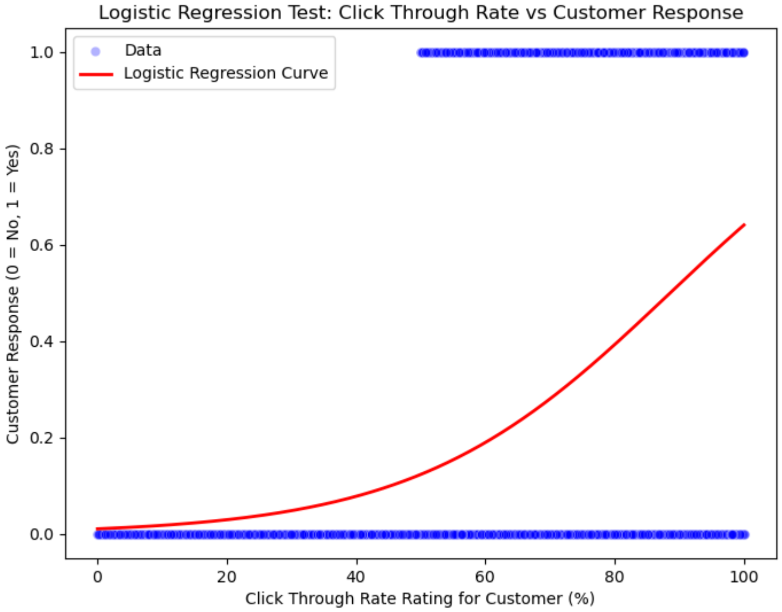


Figure 3: Logistic Regression used on response rate and click through rate for customer

Statistical Conclusion

The analysis suggests that there is a statistically significant positive relationship between the click-through rate and the likelihood of a positive response to the marketing campaign. Specifically, an increase in the click-through rate per customer is associated with a higher probability of a positive response, indicating that marketing campaigns are more likely to succeed when a higher percentage of customers engage with the ad.

Conclusion

This project examined the use of hypothesis testing, Z tests, and A/B tests to take marketing campaign related data out of the dataset and examine if the second half of campaigns (11-19) performed better than the first half of campaigns (1-9) with a one month buffer in between. This test showed that there was statistical significance in an increase in response rates between the two campaigns and showcase the ability of A/B tests to inform data-driven decision making in marketing departments.

Additionally, we used a logistic regression test to examine the relationship between customer response rate and individual click through ratings. This found that as a user's click through rating rose in percentage, so did the probability of a positive response to the marketing campaign.

By using these data analysis techniques, we can make informed decisions about what marketing campaigns are successful, and what campaigns are not. Additionally, we can better understand our customer and what customer metrics are best to focus on during marketing campaigns.

Other Directions:

In future research, we could explore the following directions:

- Use other types of data in logistic regression model
- Other descriptive, inferential, or predictive statistical analysis techniques to draw further conclusions
- Other datasets that contain real data, or larger datasets
- Machine learning models for feature detections within the dataset

References

1. E. Weed, (2021). Learning Statistics with Python, URL: [https://ethanweed.github.io/pythonbook/landingpage.html]

2. F. Quin, D. Weyns, M. Galster, and C. C. Silva, (2024). "A/B Testing: A Systematic Literature Review," Journal of Systems and Software, vol. 211, p. 112011

3. LaTeX Project, (2024). LaTeX Documentation, URL: [https://www.latex-project.org/help/documentation/]

4. Python Software Foundation, (2024). Python 3 Documentation, URL: [https://docs.python.org/3/]

5. Ziya07, (2025). Cloud-Enabled Marketing Strategy Dataset, Kaggle, URL: [https://www.kaggle.com/datasets/ziya07/cloud-enabled-marketing-strategy-dataset.]