Contents

1 Executive Summary

In this notebook, I approach the problem of performing A/B tests and intrepreating their finidngs without considering the business needs. More precisely, I consider a scnerio in which we could have a totaly different interpretaion of the result generated out of an A/B test.

2 Introduction

To depict how to align the results of our A/B tests with our business needs, I will perform an A/B test on the collected data out of a business and I try to correctly justify the result of my analysis and guide the business which action to take.

3 What is A/B Testing

A/B testing, at its most basic, is a way to compare two versions of something to figure out which performs better. While it's most often associated with marketing analysis of the websites and apps, the method itself is almost 100 years old.

4 Project

A Product Manager shared a business related dataset which was gathered out of an A/B Test that he has been conducting. I need to advise him on the result of A/B Test. the Product Manager infomed me that he ran the A/B test with three different groups: A, B, and C. Furthermore, he mentioned that the data is related to the purchasing interest of the applicants in those three groups asked me to run a test on the significant difference among them.

4.1 Testing for Significant Difference

4.1.1 Importing th erequired libraries

In [123]:
```
import pandas as pd
na_values = ['NaN']
#pd.io.parsers.read_csv('Desktop/Ross/dataset.csv', na_values=na_values, dtype={'Zip': 'str'})
df = pd.io.parsers.read_csv('best_price_point_to_offer.csv', na_values=na_values)
df.head()
```

Out[123]:

	user_id	group	click_day
0	8e27bf9a-5b6e-41ed-801a-a59979c0ca98	A	NaN
1	eb89e6f0-e682-4f79-99b1-161cc1c096f1	A	NaN
2	7119106a-7a95-417b-8c4c-092c12ee5ef7	A	NaN
3	e53781ff-ff7a-4fcd-af1a-adba02b2b954	A	NaN
4	02d48cf1-1ae6-40b3-9d8b-8208884a0904	A	Saturday

In [17]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4998 entries, 0 to 4997
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   user_id    4998 non-null   object
 1   group      4998 non-null   object
 2   click_day  582 non-null    object
dtypes: object(3)
memory usage: 117.3+ KB
```

4.1.2 Feature Enginnering

In [112]:
```
df.click_day.unique()
```

Out[112]:
```
array([nan, 'Saturday', 'Thursday', 'Friday', 'Wednesday', 'Tuesday',
       'Monday', 'Sunday'], dtype=object)
```

In [124]:
```
df['is_purchase']="value"
```

In [126]:
```
ls = ['Saturday', 'Thursday', 'Friday', 'Wednesday', 'Tuesday',
       'Monday', 'Sunday']
for row in df.index:
    if df['click_day'][row] in ls:
        df['is_purchase'][row] = 'Purchase'
    else:
        df['is_purchase'][row] = 'No Purchase'
df.head()
```

Out[126]:

	user_id	group	click_day	is_purchase
0	8e27bf9a-5b6e-41ed-801a-a59979c0ca98	A	NaN	No Purchase
1	eb89e6f0-e682-4f79-99b1-161cc1c096f1	A	NaN	No Purchase
2	7119106a-7a95-417b-8c4c-092c12ee5ef7	A	NaN	No Purchase
3	e53781ff-ff7a-4fcd-af1a-adba02b2b954	A	NaN	No Purchase
4	02d48cf1-1ae6-40b3-9d8b-8208884a0904	A	Saturday	Purchase

In [127]:
```
purchase_counts = df.groupby(['group', 'is_purchase'])['user_id'].count()
contingency = [[purchase_counts[group, 'Purchase'], purchase_counts[group, 'No Purchase']] for group in ('A', 'B', 'C')]
contingency
```

Out[127]:
```
[[316, 1350], [183, 1483], [83, 1583]]
```

Contents

I decided to count the number of purchase and not purchase desires among each group seperately.

```
contingency = [[# of groupA_purchases, # of groupA_not_purchases],
               [# of groupB_purchases, # of groupB_not_purchases],
               [# of groupC_purchases, # of groupC_not_purchases]]
```

It seems clients have shown more more purchasing behavior in Class A, I decided to perform a chi-squared test.

4.1.3 chi-squared test of independence

The following assumptions need to be meet in order for the results of the Chi-square test to be trusted.

(1) When testing the data, the cells should be frequencies or counts of cases and not percentages. It is okay to convert to percentages after testing the data
(2) The levels (categories) of the variables being tested are mutually exclusive
(3) Each participant contributes to only one cell within the Chi-square table
(4) The groups being tested must be independent
(5) The value of expected cells should be greater than 5

In [129]:
```
from scipy.stats import chi2_contingency
```

In [132]:
```
chi2, pval, dof, expected = chi2_contingency(contingency)
print ('Chi-Squared statistic:', chi2)
print ('p-value: ', pval)
print ('Degrees of Freedom: ', dof)
print ('Expectation: ', expected)
```

```
Chi-Squared statistic: 159.41952879874498
p-value:  2.4126213546684264e-35
Degrees of Freedom:  2
Expectation:  [[ 194. 1472.]
 [ 194. 1472.]
 [ 194. 1472.]]
```

The first value (159.4) is the Chi-square value, followed by the p-value (2.41e-35), then comes the degrees of freedom (2), and lastly it outputs the expected frequencies as an array. Since all of the expected frequencies are greater than 5, the chi2 test results can be trusted. We can reject the null hypothesis as the p-value is less than 0.05.

Now we need to ask the manager what why we grouped the clients in two 3 groups and what was the business motifs. According to the manager we have a newly added feature to an item for sale and the new price of the item should be defined based on the value of the new feature according to the customer point of view. But since the manager could not estimate what could be the willingness to pay of the clients, he decided to capture the opinion of the clients with 3 suggested sales plans. The Price offered to class A=0.99, B=1.99, C=4.99.

According to our obtained result it looks like more people are after the price tag = 0.99 but should we really choose this price as so?

4.2 Following the Business Goal

What we really want to know here is if each price point allows us to make enough money that we can exceed some target goal. so we should ask the manager how much do you think it cost to build this new feature in the item?

The response can be something like *"we need to generate a minimum of $1000 per week in order to justify this project"*.

In [133]:
```
#Let us look at the number of visitors
visitors = len(df)
print(visitors)
```

```
4998
```

This is the number of visitors came to the site this week. Now we need to calculate the number of people who the business would need to purchase the feature (intention to purchase for 0.99) in order to generate 1000. Then we divide it by the number of people who visit the site each week.

In [142]:
```
# For Class A
target_goal = 1000
customers_Class_A = target_goal/.99
A_percentage = customers_Class_A /visitors
print('The minimum # of weekly customers needed to purchase it for $0.99: {:.2f}'
      .format(customers_Class_A))
print('That asccounts for {:.2f}% of the weekly visitors'.format(A_percentage * 100))
```

```
The minimum # of weekly customers needed to purchase it for $0.99: 1010.10
That asccounts for 20.21% of the weekly visitors
```

In [143]:
```
# For Class B
target_goal = 1000
customers_Class_B = target_goal/1.99
B_percentage = customers_Class_B/visitors
print('The minimum # of weekly customers needed to purchase it for $1.99: {:.2f}'
      .format(customers_Class_B))
print('That asccounts for {:.2f}% of the weekly visitors'.format(B_percentage * 100))
```

```
The minimum # of weekly customers needed to purchase it for $1.99: 502.51
That asccounts for 10.05% of the weekly visitors
```

In [144]:
```
# For Class C
target_goal = 1000
customers_Class_C = target_goal/4.99
C_percentage = customers_Class_C/visitors
print('The minimum # of weekly customers needed to purchase it for $4.99: {:.2f}'
      .format(customers_Class_C))
print('That asccounts for {:.2f}% of the weekly visitors'.format(C_percentage * 100))
```

```
The minimum # of weekly customers needed to purchase it for $4.99: 200.40
That asccounts for 4.01% of the weekly visitors
```

Note that we need a smaller percentage of purchases for higher price points. 4% < 10% < 20%

4.2.1 Binomial Test

Contents

4.2.1.1 Importing th erequired libraries

In [138]:
```
from scipy.stats import binom_test
```

In [146]:
```
purchase_counts
```

Out[146]:
```
group  is_purchase
A      No Purchase    1350
       Purchase        316
B      No Purchase    1483
       Purchase        183
C      No Purchase    1583
       Purchase         83
Name: user_id, dtype: int64
```

4.2.1.2 Calculating the P value

In [151]:
```
# Test group A here
p_value_classA = binom_test(x = purchase_counts['A', 'Purchase'], n = purchase_counts['A'].sum(), p = A_percentage)
print('p-value for class A at the $0.99 price point: {:.4f}'.format(p_value_classA))
```

```
p-value for class A at the $0.99 price point: 0.2111
```

In [152]:
```
# Test group B here
p_value_classB = binom_test(x = purchase_counts['B', 'Purchase'], n = purchase_counts['B'].sum(), p = B_percentage)
print('p-value for class B at the $1.99 price point: {:.4f}'.format(p_value_classB))
```

```
p-value for class B at the $1.99 price point: 0.2066
```

In [153]:
```
# Test group C here
p_value_classC = binom_test(x = purchase_counts['C', 'Purchase'], n = purchase_counts['C'].sum(), p = C_percentage)
print('p-value for class C at the $4.99 price point: {:.4f}'.format(p_value_classC))
```

```
p-value for class C at the $4.99 price point: 0.0456
```

4.3 Conclusion

If any of the classes passed the binomial test with $p < 0.05$, then we can be confident that enough people will buy the upgrade package at that price point to justify the feature. Based on the binomial test results, the p-values for class C is the only class to pass the hypothesized probability of success of p>.05. This is interpreted as our confidence level is greater than 95 percent that enough customers buy the upgrade option at the 4.99 price point to justify offering it. **The manager should choose the 4.99 price point that tested with class C.**

In [ ]: