Ad A/B Testing

Final Report

**Problem Statement**

A/B testing is, at its most basic, a way to compare two versions of something to figure out which performs better. This method of testing has been around for a long time and is an intuitive way to compare similar things. Its no wonder that A/B testing is still an important testing method used by businesses today to look at any number of metrics (eg. customer engagement, web traffic, revenue) and see which versions increase performance.

AdSmart is looking to increase customer engagement with their questionnaire. They have recorded the data on users that have seen a dummy ad and users that have seen the new ad. Which version of their ad performs better?

**Datasets**

* AdSmartABdata – A dataset from a company running an a/b test on a new advertisement to test customer engagement with their questionnaire.
  + Half of the users are given a dummy ad and the other half are given the new ad. Measure of success is if the user clicks on ‘yes’ after viewing the ad

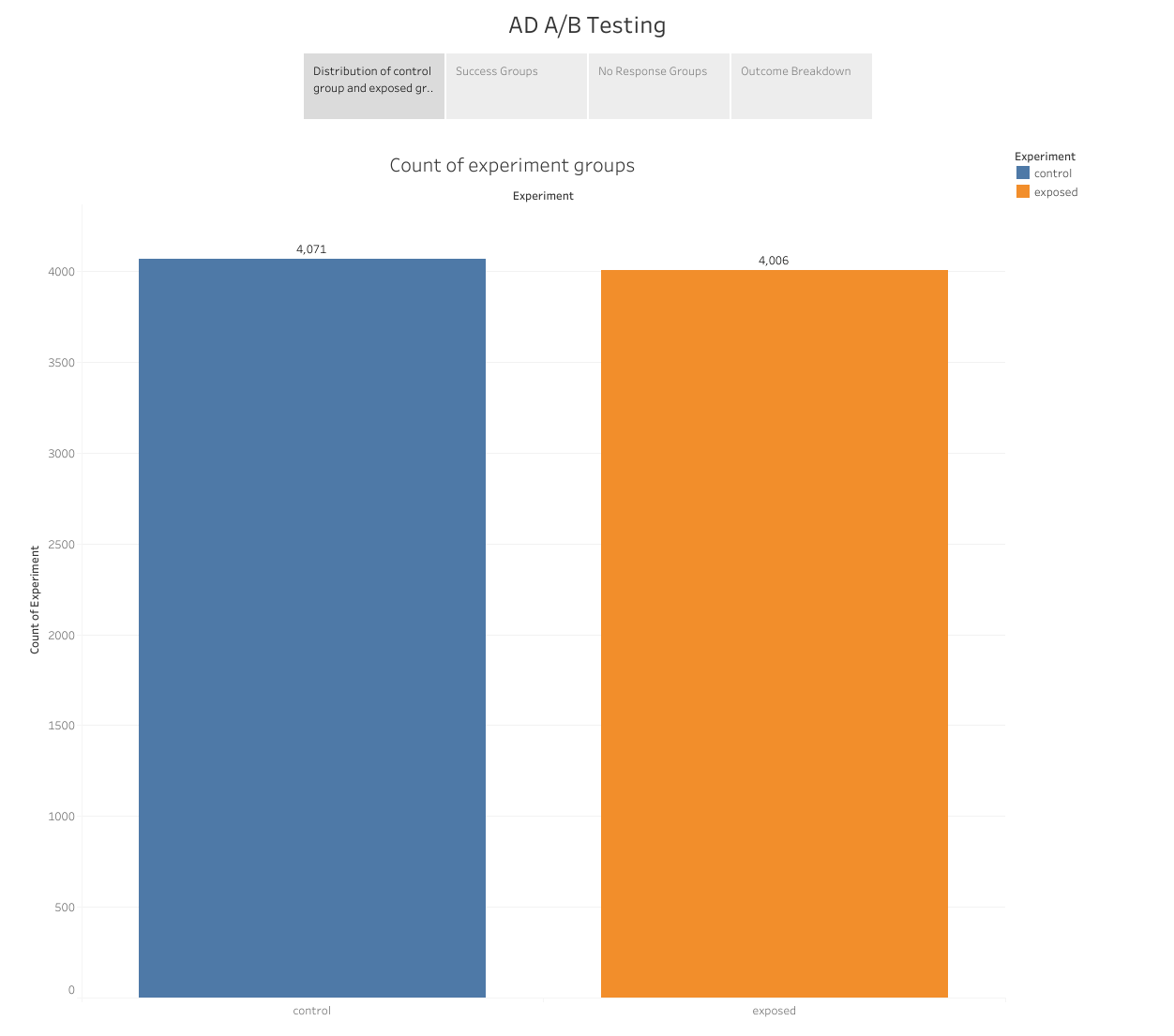
**Data Wrangling**

I took the date and hour column that were originally two separate columns and converted them to date time and combined them together. There was also a yes and no column that I changed into a success and no response column.

Success tracks if the user clicked on yes on the questionnaire at the end of the ad. No response tracks if the user did not respond to the questionnaire. Each combination of these metrics shows every possible combination of user engagement.

* if success is 0 and no\_response is 0 - user clicked on no after the ad
* if success is 1 and no\_response is 0 - user clicked on yes after the ad
* if success is 0 and no\_response is 1 - user neither clicked on yes or no
* if success is 1 and no\_response is 1 - not possible in our implementation.

**EDA**

The first step in our project was to mimic the CCi30 index. The CCi30 index takes in data from the top 30 crypto currencies and returns a value that represents the overall health of the cryptocurrency sector. It is formatted like a typical historical stock data table with “Date”, “Open”, “Close”, ”High”, ”Low”, and “Volume” as columns. Here we only use the closing price of the stock as a metric to evaluating the valuation of the stock. 

The closing price is typically used in stock market valuations as the most accurate measure of stock performance. Seen in Figure 1.

**Chart, line chart

Description automatically generated**

Figure 1: Closing Price CCi30

Google search trends all follow a similar structure. It is no surprise that specific terms such as bitcoin has the most searches. It is the largest and most popular cryptocurrency on the market.

**Chart, histogram

Description automatically generated**

Figure 2: Google Search Trends

**Model Selection**

As a precaution, I checked the correlational matrix of my independent variables (google search terms) for multicollinearity and found that while all the features are correlated with each other in some way, there were some features that were too highly correlated with each other and would affect the results of our model. We used a threshold of .80 and above as our measurement for collinearity.

Graphical user interface

Description automatically generated

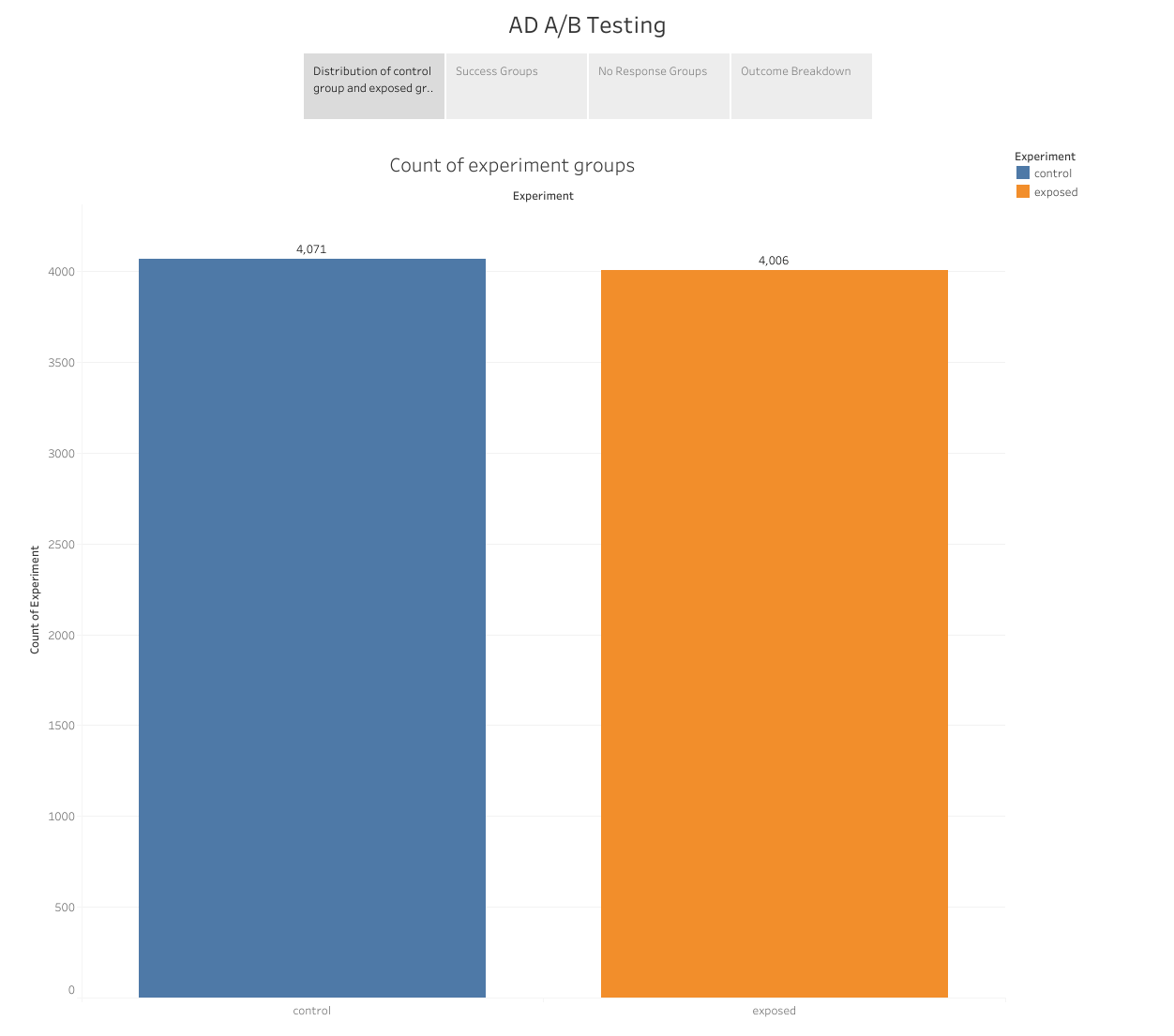
Figure 5:Correlational Matrix of my Independent Variables

To solve my issue with collinearity, I can choose to remove an independent variable and keep the other, or I can create a new column with the average of both the variables and drop both correlated variables. In this problem, I dropped the cryptocurrency google search column as it was highly correlated with multiple variables and took the average of Ethereum and Dogecoin columns.

Chart, treemap chart

Description automatically generated

Figure 6: Correlational map with collinearity fixed

For the model we tried three different regression classifiers: Linear Regression, Random Forest Regression, and Ridge Regression. We are looking for a model that can predict on our test data and has the highest r squared score with low MSE. 

Note that my data is a time series, so my cross validation must be done differently. Normal cross validation splits the train and test set randomly. You may have data that has occurred in the future in the training set, used to predict on past dates in the test set. To fix this problem, I implemented a rolling training set and test set for my data, to properly consider the time series dependency of my data.

Chart, bar chart

Description automatically generated

Figure 7:Cross Validation for time series data. (source: <https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4>)

I used the sklearn package TimeSeriesSplit to split my data 3 times and ran each model on the training set and ran a prediction on the test set. I take each r squared and MSE score from the splits and average them to find the overall r squared and MSE of each model.

**Linear Regression Results**

R squared was **0.198** for our cross-validation splits with a mean MSE of the training data of **0.142** and a mean of the test MSE of **19.32**. This is a poor model to use for this dataset due to a low score in r squared and a large number in our mean test MSE. That means that the sum of errors of our predictions is approximately 19.32 from the line of regression.

**Ridge Regression Results**

Ridge regression performed much better. R squared was lower in this model with a score of **0.11**. However, the mean MSE of the training set is a **0.16** and the mean MSE of the test set of **2.93**. I still would not choose this model, because while the lower MSE score means the predictions were more accurate, the lower r squared score means that there was a problem with how much variance our independent variables explain the dependent variable.

**Random Forest Regression Results**

R squared was much higher in this model with an average score of **0.46** for our cross validation splits with a mean MSE of the training data of **0.095** and a mean of the test MSE of **1.19**. This is our best performing model and the model I ended up choosing at the end. It has the highest r squared score and the lowest MSE of both the train and test set.

I also plotted the feature importance of our model. Bitcoin google searches has the biggest impact out of all the independent variables.

Chart

Description automatically generated

Figure 8: Feature Importance from Random Forest Regression

**Insights & Recommendations**

**Insights**

* When looking at the different statistical models, we ran the model on different metrics of 'success'
* The test for yes vs no/response shows that the differences between control and exposed group was significant. The new ad had an improved impact on customers engaging in the questionnaire
* The data shows the ad success rate of the control group to be 6.48% and the exposed group to be 7.69%
* The test for only Yes vs No responses shows that the differences were not significant.
* The test for no response rate shows that there is a significant difference between the control and exposed group. The exposed group has a lower rate of no response compared to the control group.
* The data shows the ad no response rate of the control group to be 85.61% and the exposed group to be 83.60%

**Recommendations**

* The data suggests that when looking at yes responses vs no/no response and no response rate, the new ad has more people engaging with the questionnaire.
* In the Yes vs No model, however we saw no statistical significance between the groups
* That would mean that the new ad is better at converting customers that normally would not respond
* Therefore, I would recommend the business use the new ad as it has shown to increase customer engagement with the questionnaire

**Further Research**

Further research can be done on expanding the possible variables that measure people’s interest in crypto currencies. Accessing tweets through the twitter api and looking at hashtags could provide some more insight on the type of discussion that users are generating. This can be further investigated to classify if discussions are positive or negative by using NLP algorithms.

We can also categorize people’s interest via age and gender. It would be interesting to which groups of people are talking about cryptocurrencies.

There has been an increasing amount of “scam coins” entering the market recently. These are coins that are heavily promoted via social media, luring people to invest heavily in the coin, inflating the price, only to have the owners sell all their stock and leave with the money. What if the model can measure the amount of interest a coin is generating and determine if a crypto currency is legitimate or a scam?