Team #: 9
Team Members:
1. Mary Bryan Owen (mowen32)
    Marketing Analytics - TelevisaUnivision; B.A. Claremont McKenna College
2. Edward Owen (eowen30)
    Data Scientist - Booz Allen Hamilton; B.A. Duke University
3. Fabio (fbiondolillo3)
    Data Engineer - Gilded; B.A University of Florida
4. Annie Bui (abui36)
    Business Analyst - Braze; B.S Yale University
5. Aaron (apitt6)
    Digital Data Analyst - UPS; B.S. Georgia State University

## **Project Title:** Predicting Ad Clicks

## Overview of Project

**Background Information on chosen project topic:**

In the growing digital marketplace, it is increasingly important for online retailers to reach their customers through digital ads effectively. Online marketplaces have dramatically improved the scale at which they collect user data and model it to optimize digital advertising. With this abundance of data, the possibilities to find unique patterns and model online user behavior are endless.

**Problem Statement (clear and concise statement explaining purpose of your analysis and investigation):**
Our goal is to identify the features of an ad that best predict an ad click. A marketer can leverage this knowledge to design ads incorporating these characteristics (ad features, timing, or method of display), thus improving the chances of a viewer clicking on the ad and driving profits.

**State your Primary Research Question (RQ):**
What are the key variables driving an ad click?
What are the key methodologies(models) to effectively predict ad clicks?

**Add some possible Supporting Research Questions (2-4 RQs that support problem statement):**
1. Is there a seasonal component to the probability of an ad click? (time of day, day of week, month of year)
2. What is the effect of banner position on click probability?
3. What is the effect of device type on click probability?

**Business Justification:**
This analysis will aid in predicting general online user behavior that can inform digital marketing strategies and improve return on investment from a company's marketing budget.

**DATASET/PLAN FOR DATA**
**Data Sources (links, attachments, etc.):**
We are using a dataset from Kaggle that was used for a competition:

https://www.kaggle.com/competitions/avazu-ctr-prediction/overview

***Anticipated Conclusions/Hypothesis***

Our team hypothesizes that the anonymized columns C14, C15, C16, C17, C18, C19, C20, and C21 will help predict whether or not an ad gets clicked. The consistent cleanliness of the data in these columns leads us to this conclusion. We also hypothesize Banner Position, time of day, and day of week strongly impact the likelihood of an ad getting clicked. Our approach to variable selection and the final model accuracies will allow us to confirm or reject these hypotheses.

The business impact of the results of this analysis is quite clear: They will provide marketing teams with the insight they need to optimize their advertising programs. Understanding what components of an ad campaign most heavily impact its click-through rate allows companies to cut through the noise to deliver targeted, efficient results.

## Overview of Data

**Exploratory Data Analysis**

Our dataset originally contained approximately 40 million rows. We took a random subset of 1 million rows to increase computational efficiency while providing ample training data. As a result, our data loaded into code will consist of 1 million rows and 25 columns. Each row contains information about a specific ”impression” or a single instance of a user seeing a particular ad, and each impression is assumed to be of the same advertisement. The independent variable columns are all categorical and are listed as follows:

- id
- hour
- C1
- banner_pos
- site_id
- site_domain
- site_category
- app_id
- app_domain
- app_category
- device_id
- device_ip
- device_model
- device_type
- device_conn_type
- C14
- C15
- C16
- C17
- C18
- C19
- C20
- C21

The dependent variable, “click,” will consist of binary values: 1 if the user clicked the ad upon the impression, and 0 if it was not. C1 and C14 through C21 are anonymized. The dataset provider likely did this to avoid publicly providing trade secrets that may give other companies an insight into the company in question’s competitive advantage. While this might make it difficult to derive specific business conclusions from these variables, we can still use them to demonstrate how an analyst can use any categorical variables in a similar dataset.

The “*dataprep.eda”* python package was used to begin the initial exploratory data analysis. Its results help give an initial impression of each variable and its potential usefulness in our final models. Below are some high-level statistics about the dataset:

**Dataset Statistics**

| | |
|---|---|
| Number of Variables | 25 |
| Number of Rows | $1 \times 10^{06}$ |
| Missing Cells | 0 |
| Missing Cells (%) | 0.0% |
| Duplicate Rows | 0 |
| Duplicate Rows (%) | 0.0% |
| Total Size in Memory | 739.7 MB |
| Average Row Size in Memory | 775.6 B |

A positive aspect of the dataset is that the provider has already cleaned it. The cleanliness is evident because there are zero NULL cells, which prevents us from performing any imputations or other techniques that might add bias to the original data.

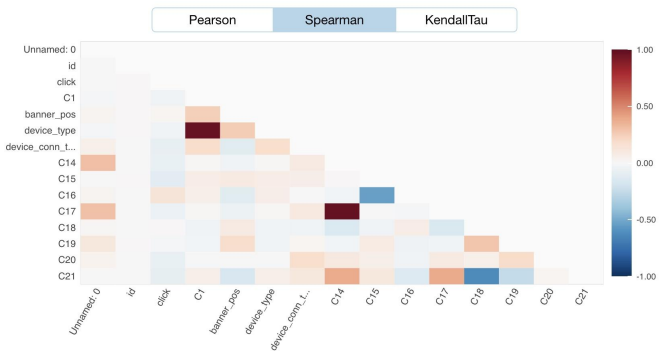
Section B of the Appendix of this paper represents a basic analysis of each variable. Each contains the counts of unique values and the total counts for each category. They provide insight into the usefulness of the categorical variables. For example, suppose a certain variable has many unique values. In that case, we can reasonably exclude it from the analysis due to the number of dummy variables it would add to the model and the lack of meaningful evidence of each value's predictive strength. The following variables can be omitted from the analysis based on their number of unique values:

- site_id: 2632 unique values
- site_domain: 2858 unique values
- app_id: 3158 unique values
- app_domain: 199 unique values
- device_id: 150228
- device_ip: 554742 unique values
- Device_model: 5174 unique values
- C14: 2258 unique values
- C17: 421 unique values
- C19: 66 unique values
- C20: 160 unique values
- C21: 60 unique values

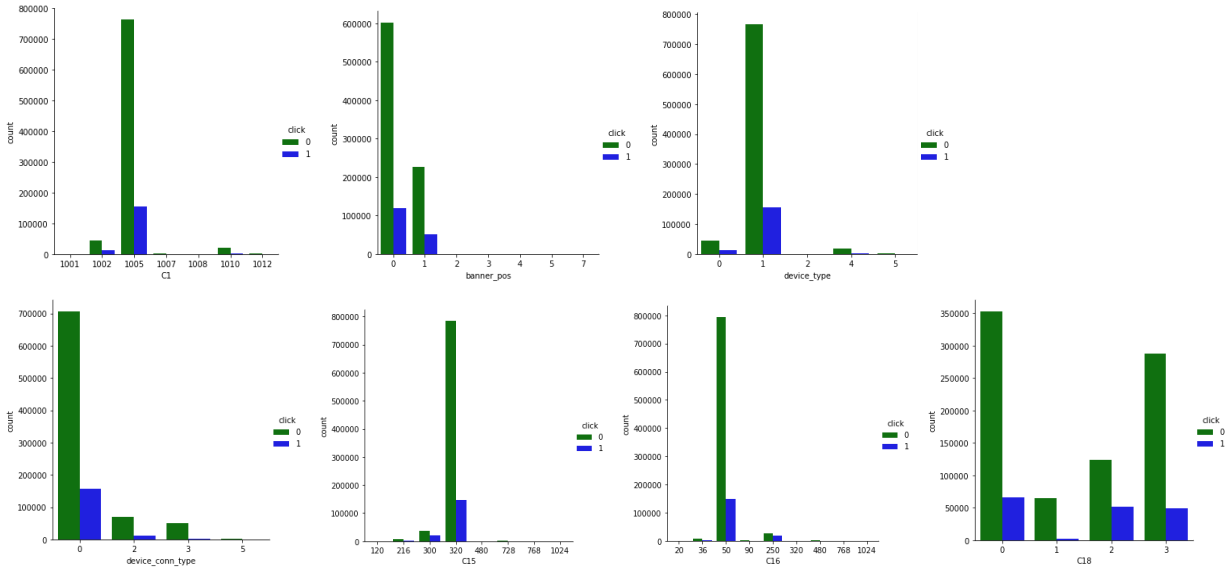
The remaining variables have a reasonable amount of unique values with which to create dummy variables, although *site_category* and *app_category* are on the higher end. They will be left in the preliminary analysis because they might have interesting results.

It is also clear from the EDA that many of these variables have the vast majority of the total count contained in 1-3 unique values. We will monitor this closely and consider the impact to ensure it doesn't introduce bias into the model conclusions.

In terms of multicollinearity, there were two main areas of concern. Still, they were related to variables we had ruled out due to category concentration and too many unique values. All other variables had an absolute value of their correlations less than .7.



To break down further the click-through rate for each of the variables above, we wanted to get a preliminary visual understanding of how that might look. The bar charts below help us to do that for the variables with fewer than ten unique values.




From the analysis results above, making any clear, immediate predictions is difficult. We can, however, hypothesize that the variables with a relatively even distribution of values will hold more predictive power in determining a click or no click.

**Data Preprocessing**

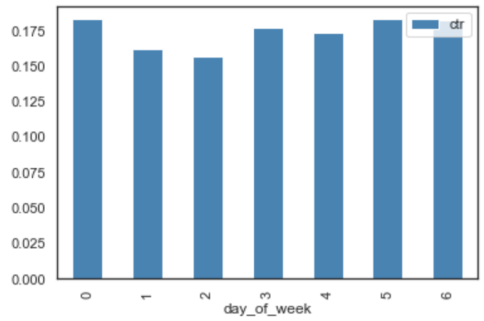
As previously mentioned, our dataset originally contained approximately 40 million rows. We took a random subset of 1 million rows and built some preliminary logistic regression models using the 1 million rows of data. We examined our dataset's poor evaluation metrics and possible connection with the class imbalance between clicks and non-clicks. There are more non-clicks (83.0%) than clicks (17.0%), which understandably could play a role in variable distribution.

We hypothesized that our logistic regression model might perform better on the majority class (non-click samples) but poorly on the minority class (click samples). We needed to address this class imbalance before building the next train, test, and validation set. If our original dataset had been small, we would have downsampled, i.e., trained on a disproportionately common subset of the majority class of non-clicks. Luckily, the original dataset provided enough records such that we didn't need to downsample or upweight the majority class. Instead, we randomly sampled 500,000 rows of clicks and 500,000 rows of non-click records.

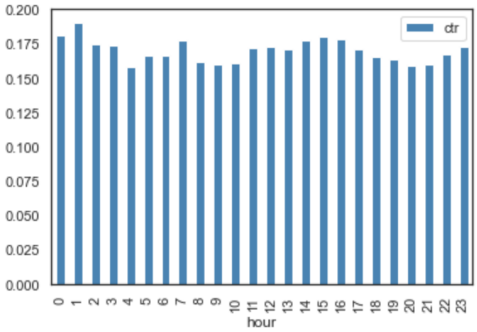
Prior to the feature engineering steps, we performed a few data preprocessing steps. First, we converted the numerical features that are categorical into factors. For example, banner position, device type, and C14 are in a numeric format but have no quantitative meaning; therefore, we wouldn't be able to perform arithmetic operations on them.

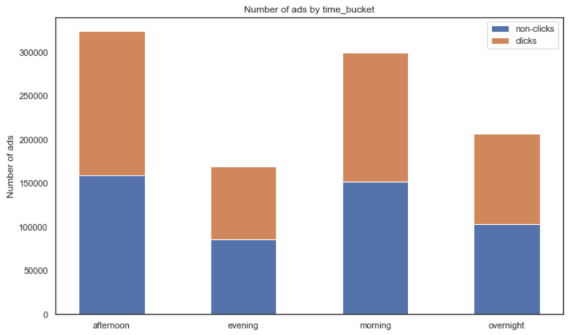
**Feature Engineering**

We experimented with timestamps to group the data into different time buckets. First, we extracted the day of the week (Monday = 0 - Sunday = 6), then the hour of the day (0-23) from the timestamps to create two new features. Then we grouped the data by day of week and hour of the day to examine changes in click-through rates across these features. When looking at the CTR for each day,




we observed some variation across days, and it appeared that Tuesday (1) and Wednesday (2) had a drop in ctr while the other days had similar ctr. When looking at the hour of the day, we observed that there were peaks and troughs in ctr throughout the day, following a somewhat cyclical pattern. Because the hour of day variable has 24 levels, we also created a new feature that grouped the hour data into four general time buckets (overnight, morning, afternoon, and evening) to see how they might influence model performance.

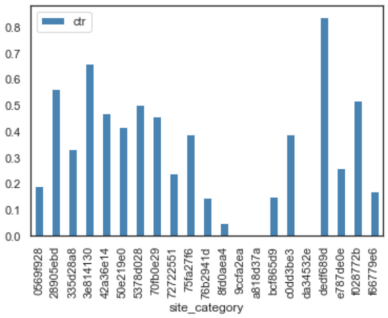


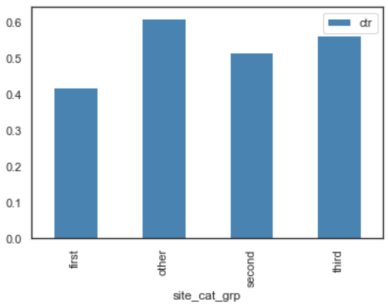




Aside from creating entirely new features from timestamp data, we experimented with binning based on the frequency of each category in some cases. The motivation behind binning was to prevent overfitting and improve the robustness of our later models. Generally, labels with low frequencies negatively affect the robustness of statistical models, and our EDA identified several categorical variables with many low-frequency values. To improve this, we created an "other" category that grouped all low-frequency values. For example, *banner position* was predominantly either a 0 or 1, accounting for 99.8% of the data. We binned the remaining *banner positions* 2-7 together.

After binning, we reduced the number of *banner position* categories from 7 to only 3. We repeated this process for several other features such as *C1* (reduced from 7 to 4)*, app category* (reduced from 27 to 6), and *site category* (reduced from 21 to 4). In addition to simply grouping low-frequency values, we also decided that we could give more meaning to our mostly anonymized dataset by using value frequency to rename some of our otherwise meaningless value names. In this case, we renamed values not in the "other" bin by the ranked order of their frequency (first, second, …., other). In the appendix, we illustrate what those features mentioned above looked like before and after binning.

*Site Category CTR With and Without Grouping*



# Overview of Modeling

### Model Overview

We split our data sample further into three groups - train (60%), test (20%), and validation (20%). We began with a basic logistic regression model using the *glm* function (see the summary output below). This model does not include feature-engineered variables and utilizes our raw, unbalanced data set. The purpose of the model is to provide us with a baseline to see how feature engineering and balancing the data into 50% clicks and 50% no clicks affects the predictive power of our logistic regression model. Based on our EDA analysis, we selected five factors (banner_pos, device_conn_type, C18, C19, C21) for our baseline model. This baseline model does not include variables that were feature engineered. We attempted to run the model for all variables, but it would crash and could not provide an output. Since the variables were hand selected, we were surprised that each was significant, as seen in the summary.

```
Call:
glm(formula = click ~ banner_pos + device_conn_type + C18 + C19 +
    C21, family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7871  -0.6583  -0.6139  -0.4509   2.6282

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.153e+00  9.654e-03 -119.43   <2e-16 ***
banner_pos        7.388e-02  6.934e-03   10.65   <2e-16 ***
device_conn_type -3.151e-01  5.245e-03  -60.08   <2e-16 ***
C18              -6.182e-02  3.129e-03  -19.75   <2e-16 ***
C19              -1.080e-04  1.008e-05  -10.71   <2e-16 ***
C21              -3.320e-03  6.438e-05  -51.57   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 546206  on 599999  degrees of freedom
Residual deviance: 538222  on 599994  degrees of freedom
AIC: 538234

Number of Fisher Scoring iterations: 4
```

We then used the test data to predict click (1) and no click (0) events. We started with a threshold of 0.5, representing the case where the model predicts a record with a probability over 0.5 as a click (1) and everything below as no click (0). We then used the validation data to calculate the accuracy of the predictions. Unfortunately, there were only predictions for the no click(0) case, meaning there are no probabilities above 50%. Through trial and error, the threshold decreased until there were also click predictions (this occurred at 0.2). While we did calculate an AUC of 0.5021, this model isn't usable because the threshold of 0.2 is unrealistic and makes it difficult to interpret.

The next model tested is similar to the logistic regression model above, except it includes feature engineering (see summary to the right). Most of the factors appear to be significant to the model. However, only *time_bucket* appears statistically significant out of the time-related factors. Like the baseline model, this model predicted 100% no clicks(0) at a threshold of 0.5, and therefore this model is usable. For comparison, we later calculated an AUC of 0.5015 at a prediction threshold of 0.2. Interestingly, the AUC decreased slightly from the baseline model in this case, showing that feature engineering had minimal effect on the model.

```
Call:
glm(formula = click ~ hour + time_bucket + hour * day_of_week +
    site_cat_grp + app_cat_grp + banner_pos_grp + device_conn_type +
    C18 + C21, family = binomial, data = train1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8943  -0.6717  -0.5524  -0.4086   2.5481

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -9.454e-01  3.397e-02 -27.829  < 2e-16 ***
hour                        -3.294e-03  2.240e-03  -1.471  0.14140
time_bucketevening          -7.766e-02  1.558e-02  -4.985 6.19e-07 ***
time_bucketmorning          -6.506e-02  1.508e-02  -4.315 1.60e-05 ***
time_bucketovernight        -8.260e-02  2.568e-02  -3.216  0.00130 **
day_of_week                  3.396e-03  4.345e-03   0.782  0.43444
site_cat_grp3e814130         4.110e-02  1.307e-02   3.144  0.00167 **
site_cat_grp50e219e0        -1.071e-01  1.581e-02  -6.774 1.25e-11 ***
site_cat_grpf028772b        -3.604e-01  1.210e-02 -29.795  < 2e-16 ***
app_cat_grp0f2161f8         -6.108e-01  1.631e-02 -37.453  < 2e-16 ***
app_cat_grp8ded1f7a         -6.832e-01  2.765e-02 -24.708  < 2e-16 ***
app_cat_grpcef3e649         -8.341e-01  2.577e-02 -32.365  < 2e-16 ***
app_cat_grpd1327cf5         -5.265e-01  4.502e-02 -11.697  < 2e-16 ***
app_cat_grpf95efa07          3.825e-01  2.240e-02  17.080  < 2e-16 ***
banner_pos_grp               8.760e-02  9.796e-03   8.942  < 2e-16 ***
device_conn_type            -1.995e-01  5.670e-03 -35.183  < 2e-16 ***
C18                         -2.124e-03  3.429e-03  -0.619  0.53562
C21                         -2.733e-03  6.564e-05 -41.635  < 2e-16 ***
hour:day_of_week            -3.351e-04  3.373e-04  -0.993  0.32048
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 546206  on 599999  degrees of freedom
Residual deviance: 531030  on 599981  degrees of freedom
AIC: 531068

Number of Fisher Scoring iterations: 5
```

Next, we switched back to a basic logistic regression model without feature engineering to see how our model performed with the balanced dataset - 50% clicks and 50% no clicks. As you may note, we included some additional factors in this model. In our code, there are two models for this section- one with the same factors from the baseline model and the one that includes additional factors. The first model had an AUC of 0.5833, which is a good improvement from our unbalanced base model. However, since many of the variables are skewed, we wondered if by balancing the data set, some of those variables might be more useful now. We found that some of them were, and our AUC increased further to 0.5882 with the prediction threshold of 0.5.

```
Call:
glm(formula = click ~ banner_pos + device_conn_type + as.factor(device_type) +
    C14 + C15 + C16 + C17 + C18 + C19 + C20 + C21, family = "binomial",
    data = train2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2430  -1.1503   0.3377   1.1496   2.1467

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                 1.462e+00  5.822e-02  25.116  < 2e-16 ***
banner_pos                  2.430e-01  6.034e-03  40.269  < 2e-16 ***
device_conn_type           -2.703e-01  3.847e-03 -70.271  < 2e-16 ***
as.factor(device_type)1    -3.837e-01  1.139e-02 -33.672  < 2e-16 ***
as.factor(device_type)4    -1.092e+00  2.744e-02 -39.787  < 2e-16 ***
as.factor(device_type)5    -1.190e+00  6.635e-02 -17.937  < 2e-16 ***
C14                        -6.709e-05  2.605e-06 -25.759  < 2e-16 ***
C15                        -1.533e-03  1.743e-04  -8.795  < 2e-16 ***
C16                         5.888e-03  5.919e-05  99.475  < 2e-16 ***
C17                         3.420e-04  2.140e-05  15.981  < 2e-16 ***
C18                        -8.697e-02  2.480e-03 -35.064  < 2e-16 ***
C19                        -2.482e-05  7.559e-06  -3.283  0.00103 **
C20                        -2.954e-06  5.383e-08 -54.882  < 2e-16 ***
C21                        -2.068e-03  5.021e-05 -41.183  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831776  on 599999  degrees of freedom
Residual deviance: 797631  on 599986  degrees of freedom
AIC: 797659

Number of Fisher Scoring iterations: 4
```

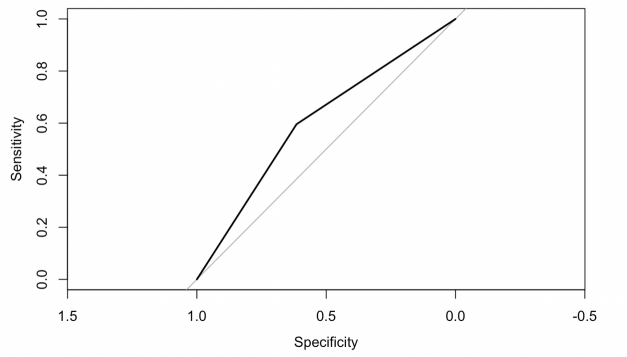
The final logistic regression model to test utilized both feature engineering and the balanced dataset. We decided to include the variables from the previous factor exploration and the featured engineered groups that performed best. While we tested many iterations of models in our code, this is the final and best model below. We removed the engineered time features because they had little effect on the tested models. With these adjustments, we again were able to predict using our test data at a threshold of 0.5 and then use our validation data in a confusion matrix with the predictions to determine accuracy. The model accuracy is 60.44%, and the AUC is 0.6052. Both an improvement from the non- feature engineered model and the unbalanced data set. (See ROC curve below)

```
Call:
glm(formula = click ~ app_cat_grp + site_cat_grp + banner_pos_grp +
    device_conn_type + as.factor(device_type) + C14 + C15 + C16 +
    C17 + C18 + C19 + C20 + C21 + C1_grp, family = binomial,
    data = train2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0828  -1.1403   0.4708   1.1146   2.1615

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -4.048e+01  7.056e+00  -5.737 9.65e-09 ***
app_cat_grpfirst            -1.674e+00  1.032e-01 -16.222  < 2e-16 ***
app_cat_grpfourth           -1.042e+00  2.220e-02 -46.940  < 2e-16 ***
app_cat_grpother            -9.595e-01  3.416e-02 -28.087  < 2e-16 ***
app_cat_grpsecond           -9.532e-01  1.616e-02 -58.985  < 2e-16 ***
app_cat_grpthird            -1.277e+00  2.150e-02 -59.385  < 2e-16 ***
site_cat_grpother            9.030e-01  1.020e-01   8.856  < 2e-16 ***
site_cat_grpsecond           9.255e-01  1.021e-01   9.064  < 2e-16 ***
site_cat_grpthird            1.350e+00  1.023e-01  13.192  < 2e-16 ***
banner_pos_grp               3.470e-01  8.784e-03  39.510  < 2e-16 ***
device_conn_type            -2.069e-01  4.080e-03 -50.707  < 2e-16 ***
as.factor(device_type)1 -1.526e+00  1.135e-01 -13.446  < 2e-16 ***
as.factor(device_type)4 -2.252e+00  1.396e-01 -16.135  < 2e-16 ***
as.factor(device_type)5 -2.361e+00  1.513e-01 -15.605  < 2e-16 ***
C14                         -7.671e-05  2.655e-06 -28.890  < 2e-16 ***
C15                         -1.690e-03  1.750e-04  -9.658  < 2e-16 ***
C16                          6.371e-03  6.765e-05  94.181  < 2e-16 ***
C17                          4.118e-04  2.178e-05  18.910  < 2e-16 ***
C18                         -2.582e-02  2.697e-03  -9.573  < 2e-16 ***
C19                          2.398e-05  7.679e-06   3.123  0.00179 **
C20                         -2.123e-06  5.525e-08 -38.419  < 2e-16 ***
C21                         -1.719e-03  5.105e-05 -33.672  < 2e-16 ***
C1_grp                       4.340e-02  7.088e-03   6.124 9.15e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

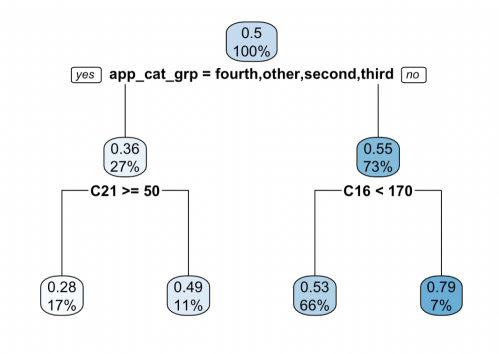



After determining the best logistic regression model based on feature engineering and balancing our dataset, we tested other model types using the factors from our final model as the inputs for future models.

First, we wanted to test two basic classification models, SVM and KNN. However, we determined that those models were not feasible- even when converting all the variables to dummy variables. The computation was too inefficient to continue tuning, and the initial results were unfavorable. As a result, we were only able to successfully run Elastic Net, ridge regression, lasso, a gradient classification tree (CatBoost), and a tree classification model(rpart).

Elastic Net, Ridge Regression, and Lasso regression are all used for feature selection or modeling. In this case, we used it for modeling. Each model has a method for penalizing extra variables to prevent overfitting. The algorithms used to pare down the features would ideally result in simpler, more explainable models. However, the performance of each model was disappointing. The following R-squared values were: Ridge- 0.054, Lasso- 0.055, ElasticNet- 0.055. In other words, only ~5% of the variance in the data can be accounted for by each model. Therefore, these models were unusable.

Next, with slightly more success, we tried CatBoost - "*a high-performance open source library for gradient boosting on decision trees*"(https://catboost.ai/). We picked this model because it natively handles categorical variables computationally efficiently. Since Catboost requires minimal feature engineering, we tested many combinations of variables, binned or otherwise, and achieved an AUC of 0.63 using only C17. To see which model ran best regardless of variable selection, we performed CatBoost on the inputs we selected in our final logistic regression, resulting in an AUC of 0.608. This AUC slightly improved from the baseline of 0.6052 but was lower than the single variable CatBoost for C17. Since CatBoost is more of a black box, we would prefer a more explainable model.




Lastly, we tested a basic tree classification model using the function “rpart.” Again, the function's inputs were the factors from the logistic regression model with feature engineering built off the balanced dataset. The model performed best with a prediction threshold of 0.6, an improvement from the final logistic regression model. Additionally, AUC was 0.6568, which is not only an improvement from the logistic regression model but also CatBoost.

**Conclusions**

In our initial hypotheses, we thought that a few key variables in our dataset would significantly impact the ability to predict whether or not a user would click an ad based on a given impression. These variables include banner position and site category from the raw data and time features we could extract from the timestamp data (day of week and hour of day). Our findings from EDA show variation among these variables, and we were able to make a few key observations from the key variables:

- Tuesdays and Wednesdays seem to have lower CTR
- There are several crests and troughs throughout the day for CTR that show cyclical variation
- Banner position has little variation in CTR for the first two positions (account for 99.8% of data)
- Site category appeared to have a significant variation in CTR across category groups

Although EDA indicated small patterns of variation for CTR data and while most models found these variables to be statistically significant, the key variables on which we formed our hypotheses did not significantly change the predictive power of any models tested. Feature engineering as a whole did not significantly improve model performance. New time features marginally improved AUC, while categorical variables with grouped rare values marginally decreased model performance.

The best model tested was an rpart tree classification model with the balanced dataset that returned an AUC of 0.6568, an improvement from an AUC of 0.5021 in our baseline model.

```
Call:
rpart(formula = click ~ app_cat_grp + site_cat_grp + banner_pos_grp +
    device_conn_type + as.factor(device_type) + C14 + C15 + C16 +
    C17 + C18 + C19 + C20 + C21 + C1_grp, data = train2, method = "anova")
  n= 600000

          CP nsplit rel error    xerror         xstd
1 0.02872874      0 1.0000000 1.0000068 3.322708e-06
2 0.01724506      1 0.9712713 0.9712799 4.265630e-04
3 0.01109540      2 0.9540262 0.9540386 5.050708e-04
4 0.01000000      3 0.9429308 0.9429450 5.521668e-04

Variable importance
          app_cat_grp           site_cat_grp                    C16                    C15
                   25                     16                     15                     12
                  C21       device_conn_type                    C17                    C14
                   10                      4                      4                      3
                  C18                    C20 as.factor(device_type)                  C1_grp
                    3                      3                      2                      1
                  C19
                    1
```

The key variables for predicting ad clicks in the best rpart model were app_cat_grp (grouped app categories), C21 (anonymous variable), and C16 (anonymous variable). None of these key variables were the initial variables that we hypothesized.

Finally, in terms of methodology, our research indicates that with typical large datasets of ad impression data, basic classification models like logistic regression appear to be weak classifiers. Balancing the dataset to oversample impressions resulting in clicks achieved the most significant model improvements. More advanced models like tree models appear to achieve much better results than models like logistic regression and KNN.

## Appendices

### Appendix A: Key Assumptions and Considerations

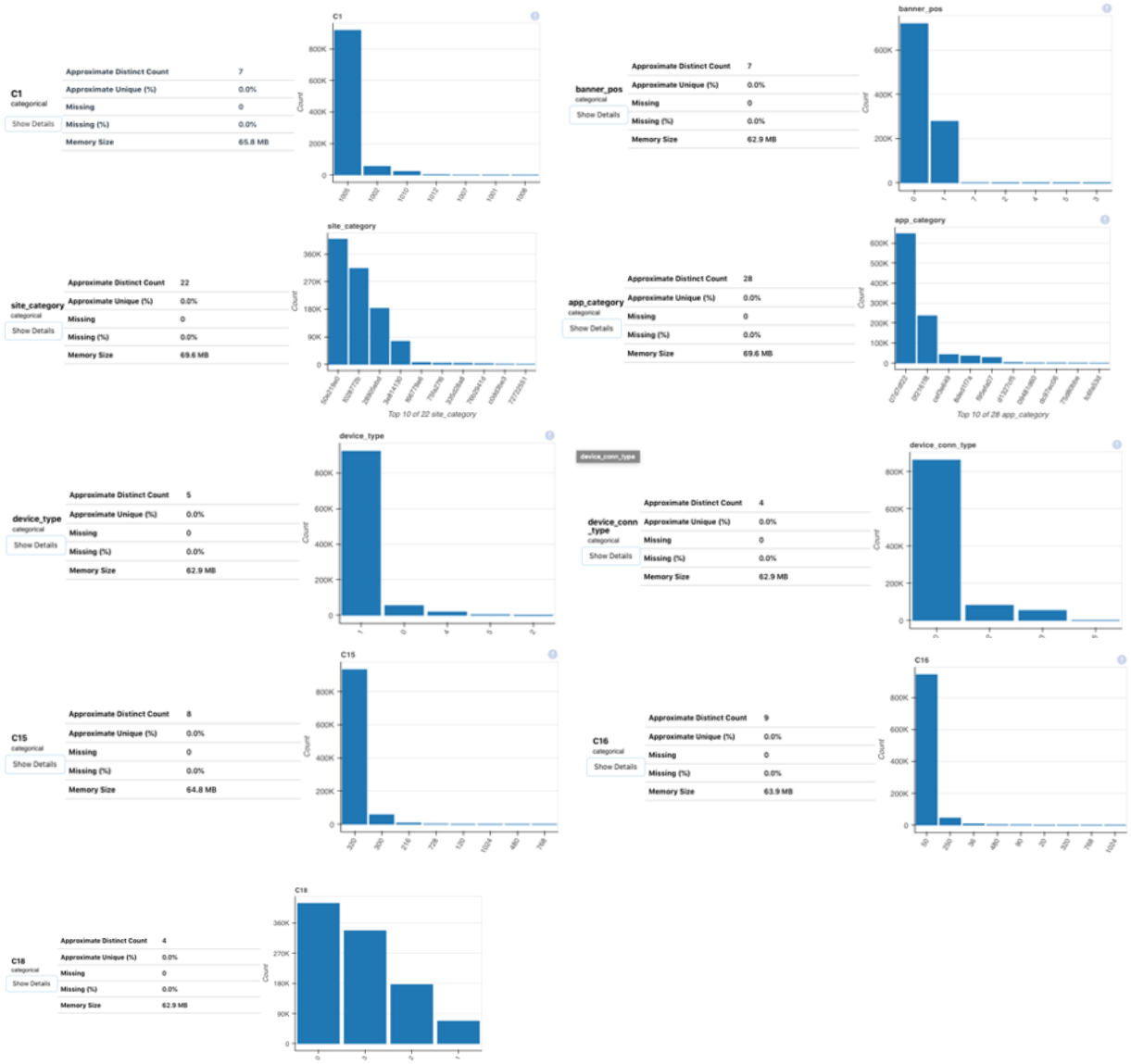
The primary assumption we are making with this dataset is that all of this anonymized data relates to the same ad and product. In other words, each impression controls for the same ad, and there are no different ads or products that could significantly influence the click-through rate.

Our group's second assumption is that the time data can be generalized to the entire year. Our dataset consists of 10 consecutive days of data. We assume that these ten days follow normal behavioral patterns, meaning that all other weeks in the year exhibit similar click behavior, and there are no special seasonal effects that this dataset doesn't capture.

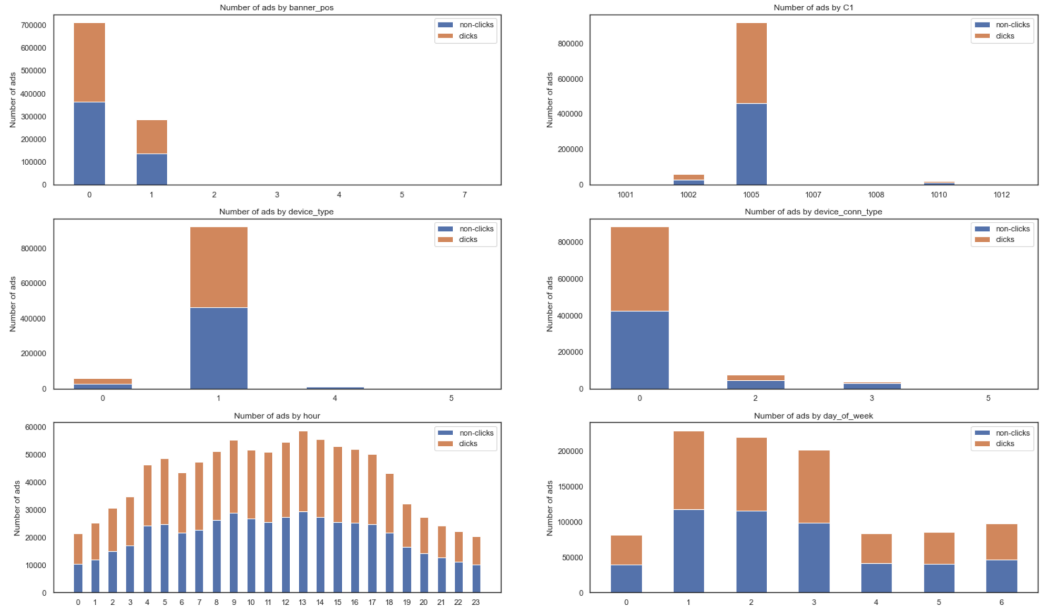
For data sampling, we assume that random selection did not sample the data in a way that would introduce bias into our training dataset. When oversampling for clicks to create a balanced training dataset, we assumed that that oversampling did not introduce any unexplained bias. In addition, when analyzing the models we tested, many of them were such weak classifiers with the imbalance dataset that we found them to be mostly unusable with the training data we selected.

Our conclusions came from a finite set of classification models we tested on the data. There could be other algorithms that perform much better than those we selected that exist outside of what we tested in our analysis.

### Appendix B: Automated EDA charts



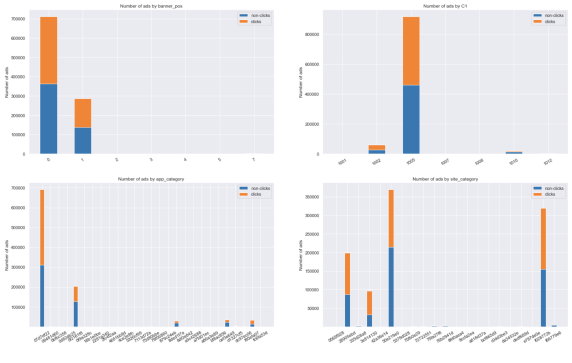
## Appendix C: EDA for categories and time categories broken by Clicks and Non-Clicks



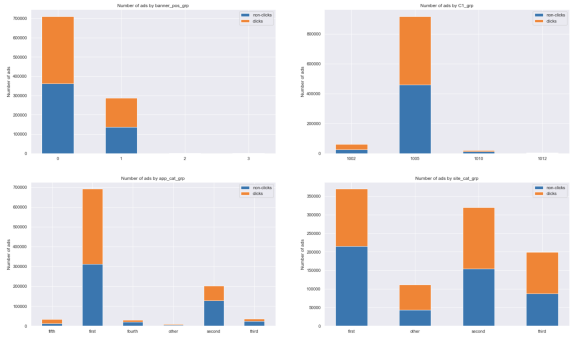

Time bucket categorization:

| | Time Start | Time End |
|---|---|---|
| **overnight** | 00:00 | 06:00 |
| **morning** | 06:00 | 12:00 |
| **afternoon** | 12:00 | 18:00 |
| **evening** | 18:00 | 24:00 |

Before Binning:



After Binning:



**Works Cited**

Kaggle Dataset:
https://www.kaggle.com/competitions/avazu-ctr-prediction/overview
Open Benchmarking for Click-Through Rate Prediction:
https://arxiv.org/pdf/2009.05794.pdf
Click-Through Rate Prediction in Online Advertising: A Literature Review:
https://arxiv.org/pdf/2202.10462.pdf
Ad Click Prediction: a View from the Trenches:
https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41159.pdf
CTR Benchmarking:
https://arxiv.org/pdf/2009.05794.pdf
Factorization machine:
https://docs.aws.amazon.com/sagemaker/latest/dg/fact-machines.html
CatBoost:
https://catboost.ai/en/docs/concepts/fstr