DATASCI 450: Deriving Knowledge from Data at Scale

Team 3-Trees

Todd Schultz, et al

Contents

Initial Data Exploration

Data Size

- It's a huge training dataset: 5.87 GB comma separated values text file
- 40,428,967 observations with 24 columns!
- Training set = 10 days of data
- Most personal computers cannot handle it, need sampling

Data Quality

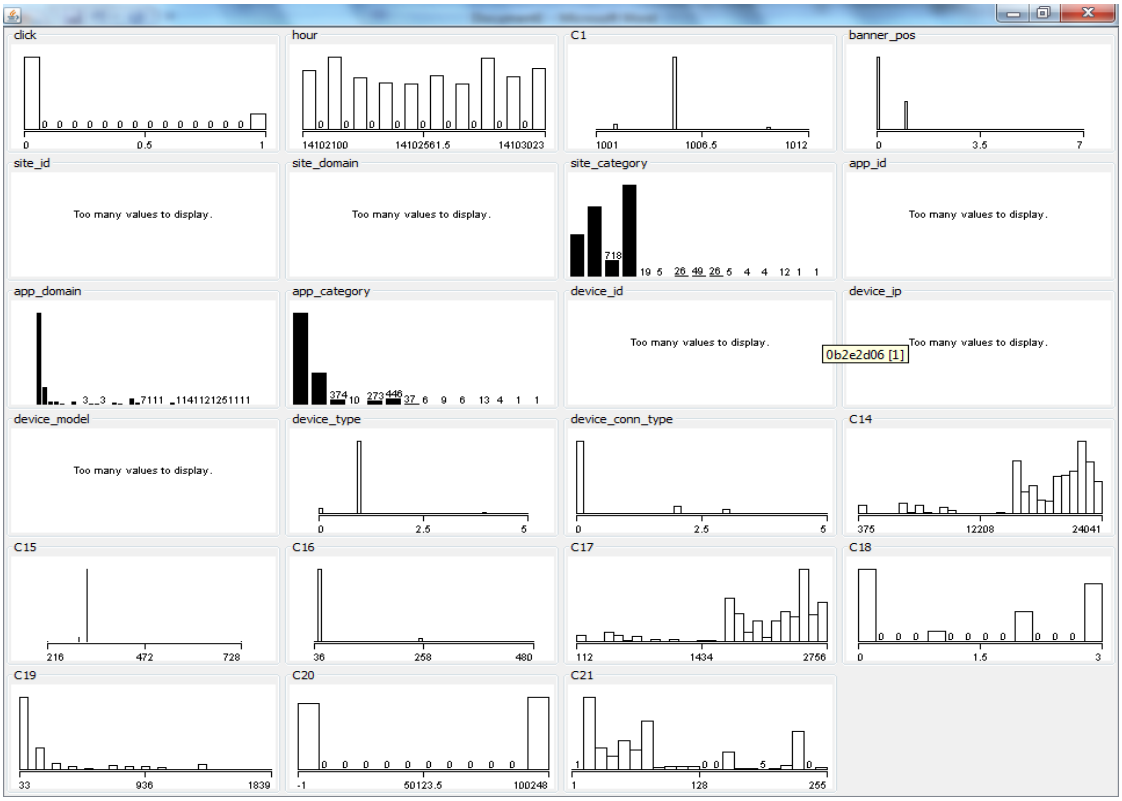
1. All variables are categorical
    a. Supervised discretization an option for feature extraction
2. Lack of definition on most variables
    a. Sample values such as f3845767
    b. No units or descriptions, could not make sense either thru variable names (half of'em) or thru the observations from any of the variables
    c. No information on bounds, ranges
3. Hour
    a. Portion of data that clicked is constant for each hour
    b. In UTC, not local time
4. The variable "Id" found to be surrogate key for the data set, i.e. its having 100% distinct values hence concluded as proxy for building model
5. High amount of distinct values/levels found in some of the variables too, can't think of a way to bin/discretize them into smaller categories

Dealing with Missing Values

Variables are obfuscated, even NULL & missing values. There is no opportunity to deal with missing values.

Data Scaling

All values are categorical. There is no opportunity to normalize or scale the data.



Resampling

Original Dataset

The original training dataset provide by Kaggle is a 5.87 GB csv file with a total of 40,428,967 observations and 24 columns from 10/21/2014 12:00 am UTC to 10/30/2014 11:00 pm UTC. This file is deemed too large to work on most computers and thus will be resampled to reduce its size.

- The 24 columns are id, click, 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, and C21.
- All predictor variables are categorical.
- The hour variable is the time code segmented into hour long segments written as YYMMDDHH. Just on a hunch, the 'hour' variable was grouped into 24 unique levels, one for each hour of the day and the correlation recomputed. The correlation coefficient between clicks and this new hour variable was -0.0015.
- The number of unique categories for each variable is given below along with the naïve estimate of the correlation coefficient with the click (response) variable. Nothing truly stands out as a 'must have' variable so more work is needed.

Table: Correlation coefficients with respect to the variable 'click' for the entire dataset provide by Kaggle.

Variable	Number of unique categories	Correlation Coefficient with click
'id'	-	-
'click'	2	1
'hour'	240	-0.00778
'C1'	7	-0.03525
'banner_pos'	7	0.025535
'site_id'	4,737	-0.01604

'site_domain'	7,745	-0.02951
'site_category'	26	-0.01854
'app_id'	8,552	-0.04409
'app_domain'	559	-0.03002
'app_category'	36	-0.04555
'device_id'	2,686,408	-
'device_ip'	6,729,486	-
'device_model'	8,251	-0.00359
'device_type'	5	-0.03776
'device_conn_type'	4	-0.08636
'C14'	2,626	-0.06113
'C15'	8	-0.0946
'C16'	9	-0.13744
'C17'	435	-0.06197
'C18'	4	0.021634
'C19'	68	-0.03157
'C20'	172	-0.07076
'C21'	60	-0.07104

To understand the population of the original data better before resampling, the proportions of the populations for two variables was investigated. Out of the entire 40,428,967 observations, only 6,865,066 observations clicked on the ad for a percentage of 16.98 %.

Next, I determined that the number of observations for each hour long segment was not uniform. Both the percentage that clicked and the percentage in each hour long segment would have to be accounted for in the resampling strategy. The last thing considered before resampling was the percentage that clicked on the ad for each hour of the day. Again, here the hour variable was grouped into 24 segments, one for each hour of the day. The figure below shows the breakdown of the percentage that clicked for each hour of the day. The percentage of observations that clicked the ad for each hour of the day is uniform to within 3%, thus will be considered uniform. This leads me to believe that either the original dataset was sampled to enforce equal percentages or the hour variable will not be an important predictor.

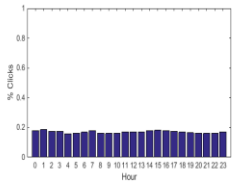


Figure: Percentage of observations for each hour of the day that clicked on the ad.

Resampled Dataset

The decision to resample the data and reduce its size was made for practical reasons. The original dataset provided by Kaggle is simply too large for most consumer grade (laptop) computers to handle in a time efficient manner. If I was working on this data as a business venture, I would look for computational resources to handle the data size such as a workstation class desktop computer or cloud resources. Here, resampling will increase the productivity of our team by allowing the use of their home, consumer grade computers.

Sampling Strategies - preserves the percentages of those that clicked the ad and the percentages of data from each hour long segment

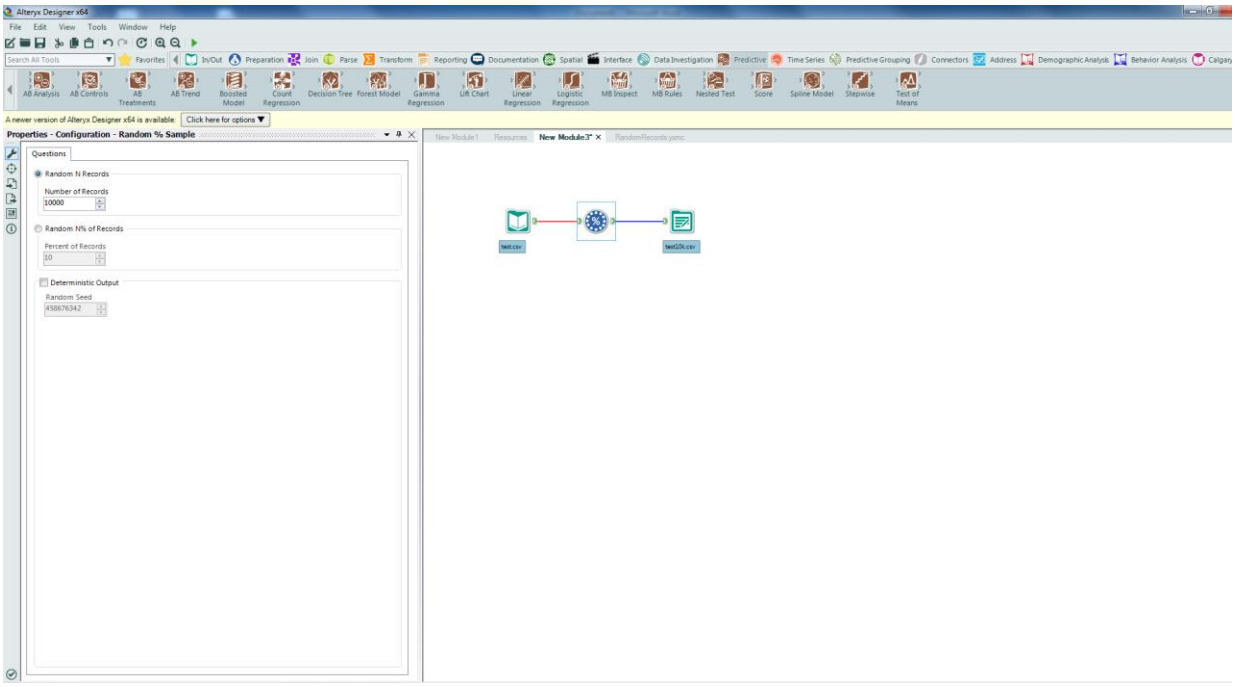
I generated two sample dataset with the same strategy, but with varied amounts of data. The smaller dataset was designed to have 100,000 samples and the second one was designed to have 500,000 samples. Both were also provided with a smaller, independent validation dataset so that the entire training set may be used for training the model. The sampling strategy was a two layered stratified sampling technique that preserves the percentages of those that clicked the ad and the percentages of data from each hour long segment, all 240 of them. The major steps are outlined below.

1. Determine overall percentage of clicks and data in each hour segment.
2. Initialize new training dataset with the first row of the original data and the validation dataset with the second row.
3. Pull in the next group of observations and the total number of observations are present and determine how many hour long segments are present.
4. Break the data sample into each hour long segment and then subdivide each hour long segment into those that clicked and those that didn't click.
5. Pull a random sample without replacement proportional to the total number of samples desire to the number of observations in each subsample and save as the training set.
6. Remove all samples that have been saved into the training set from the subsample groups.
7. Pull another random sample without replacement proportional to the number of samples desires for the validation set and save.
8. Repeat for the clicked and no clicked data.
9. Repeat for each hour long segment.
10. Repeat until the end the original data file from Kaggle has been reached.

The new reduced size samples were then checked for consistency with the percentage of clicks and with the population distribution of data in each hour long segment and were found to be identical with regards to those two statistics. Now the reduced size datasets can be used for machine learning. The 500k sample contained 500,048 observations in the training set and 99.949 observations in the validation set. The minor difference in the final observation counts compared to the desired values were due to rounding to integer values for each data pull from the original file. The 100k sample contained 99,949 observations for the training set and 50,049 for the validation set.

Other Sampling Strategies

- Using commercial software:
  This is the lazy and frustrated analyst approach but Alteryx is also the tool of choice for $300+/hour outside consultant. Using their random sample module, two sets of data were compared: 10K vs 100K. Value distribution was consistent between 10K and 100K so 10K was used for this exercise. It took <10 minutes to get the samples out. After realizing that 50/50 sample does not work, 83/17 was used. Below is a print-screen of Alteryx random module + their predictive analysis modules.


Alteryx Designer x64
File Edit View Tools Window Help
Properties - Configuration - Random % Sample
Questions
Random N Records
Number of Records
10000
Random N% of Records
Percent of Records
50
Deterministic Output
Random Seed
458676342
New Module3*
test.csv
test10k.csv

Feature Selection

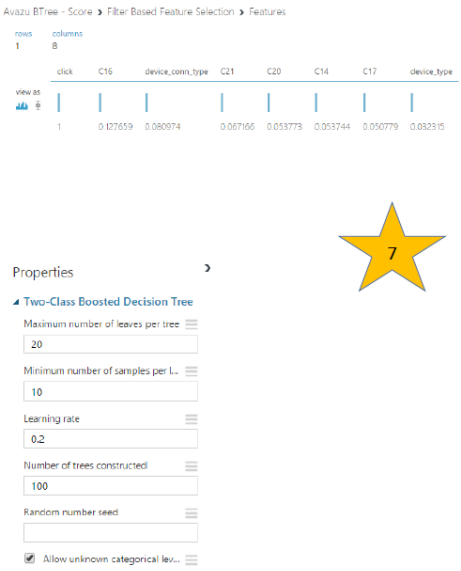
A number of methods were used to select features for each of the ensemble models:

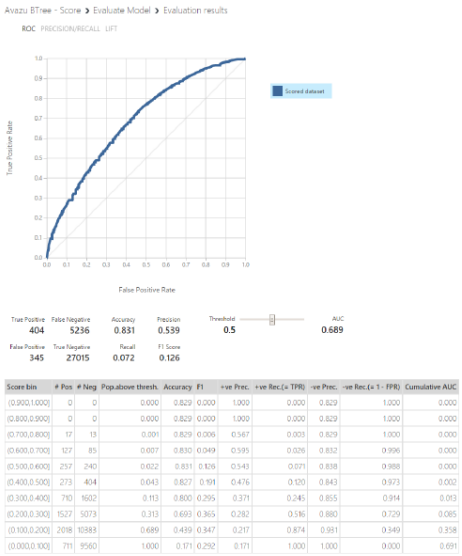
Azure ML - 7-Variable Boosted Decision Tree

In the 7-Variable Boosted Decision Tree models developed in Azure ML, an iterative feature selection approach was used to identify the attributes that most contributed to a predictive model without necessarily over fitting to the data. Azure ML's "Filter Based Feature Selection" module using the Pearson Correlation method was applied with a range of "desired features" settings. The resulting features were then iteratively modeled to balance F-score and AUC against the complexity of the model.

Desired Features	*F-score*	*AUC*
10	0.128	0.692
9	0.128	0.690
8	0.126	0.689
7*	***0.126***	***0.689***
6	0.118	0.683
5	0.116	0.682
4	0.056	0.653
3	0.089	0.619

The 7-variable filter was determined to best balance complexity against predictive power, resulting in the following ROC chart:





Weka

A few attempts were made using Weka -

- Used Weka's Select Attributes tab, tested OneR, Chi-Squared ranking, Gain Ratio, Information Gain, correlation ranking filter. Select the highest accuracy model using the different attribute sets. Select top attributes that show up on multiple Select Attributes modules and test on different models.

- Ran naïve bayes and bagging options on best first search method on sample set of 10k on training set, out of all the suggested ones by the tool, below are the most common features I have selected: site_category, device_model, C14

- Used Weka to select ranked attributes with 4 different evaluators, (Correlation, InfoGain, Gain ratio, Chi squared) and then chose the intersect of top 14 attributes from each evaluator result. An heuristic approach was employed to try different combinations of attributes and determine that 14 attributes is an optimal pool size for selecting such features.

Ranked attributes:	Ranked attributes:	Ranked attributes:	Ranked attributes:
0.118415 11 device_model	0.032448 16 C16	1552.602978 11 device_model	0.12567 16 C16
0.114766 14 C14	0.024405 15 C15	1531.009436 14 C14	0.11756 15 C15
0.086263 5 site_id	0.018385 5 site_id	1258.877251 5 site_id	0.09798 10 app_category
0.076281 6 site_domain	0.017554 6 site_domain	1111.432619 6 site_domain	0.07954 9 app_domain
0.074386 17 C17	0.015751 8 app_id	974.359287 17 C17	0.07661 8 app_id
0.05027 8 app_id	0.014339 11 device_model	576.973507 8 app_id	0.06796 13 device_conn_type
0.034396 21 C21	0.014037 14 C14	426.000905 21 C21	0.0617 6 site_domain
0.029063 19 C19	0.013021 18 C18	360.870633 19 C19	0.06126 5 site_id
0.025411 20 C20	0.011157 17 C17	302.116952 18 C18	0.05836 18 C18
0.022713 18 C18	0.009891 9 app_domain	268.212902 20 C20	0.05671 7 site_category
0.018913 9 app_domain	0.009108 10 app_category	213.094516 2 hour	0.03897 20 C20
0.01607 2 hour	0.008373 13 device_conn_type	206.724627 9 app_domain	0.03752 21 C21
0.01358 10 app_category	0.00823 21 C21	191.008285 16 C16	0.03494 19 C19
0.010996 16 C16	0.007676 19 C19	174.342039 10 app_category	0.02384 17 C17
0.010607 7 site_category	0.006981 20 C20	153.771001 15 C15	0.01669 14 C14
0.009217 15 C15	0.005541 7 site_category	144.803172 7 site_category	0.01635 4 banner_pos
0.006017 13 device_conn_type	0.003632 3 C1	67.039429 13 device_conn_type	0.01608 12 device_type
0.00186 3 C1	0.00339 12 device_type	23.490734 3 C1	0.01423 11 device_model
0.00163 12 device_type	0.002072 2 hour	21.682076 12 device_type	0.0137 3 C1
0.000282 4 banner_pos	0.000328 4 banner_pos	3.424369 4 banner_pos	0.0079 2 hour

- We used ensemble technique to select attributes. We created two data sets with 100K rows using python tool called subsample (https://pypi.python.org/pypi/subsample/0.0.6) and then used the following technique.

    - CfsSubsetEval + BestFirst
    - CfsSubsetEval + Random Search
    - Infogain + Ranker
    - Chisquared +Ranker
    - Gainratio + Ranker
    - Classifier + Ranker

Train_Data_Set_1		
CfsSubsetEval + BestFirst	**CfsSubsetEval + Random Search**	**Infogain + Ranker**
10(100 %) 3 site_domain	10(100 %) 5 site_domain	0.064 +- 0 1 +- 0 10 C14
1(10 %) 5 app_domain	9(90 %) 7 app_domain	0.051 +- 0 2 +- 0 3 site_domain
10(100 %) 9 device_conn_type	1(10 %) 9 device_model	0.05 +- 0 3 +- 0 13 C17
1(10 %) 11 C15	1(10 %) 11 device_conn_type	0.049 +- 0 4 +- 0 7 device_model
10(100 %) 12 C16	1(10 %) 13 C15	0.031 +- 0 5 +- 0 17 C21
10(100 %) 14 C18	10(100 %) 14 C16	0.026 +- 0 6 +- 0 15 C19

	10(100 %) 16 C18	0.022 +- 0 7 +- 0 14 C18
	1(10 %) 17 C19	

Train_Data_Set_1	
Chisquared +Ranker	**Gainratio + Ranker**
7808.03 +-37.73 1 +- 0 12 C14	0.033 +- 0 1 +- 0 12 C16
6731.958 +-50.143 2 +- 0 5 site_domain	0.024 +- 0 2 +- 0 11 C15
6232.856 +-42.301 3 +- 0 15 C17	0.013 +- 0 3 +- 0 14 C18
5694.353 +-29.392 4 +- 0 9 device_model	0.012 +- 0 4 +- 0 3 site_domain
3544.321 +-32.744 5 +- 0 19 C21	0.011 +- 0 5 +- 0 9 device_conn_type
2893.334 +-35.557 6 +- 0 17 C19	0.009 +- 0 6 +- 0 5 app_domain
2644.203 +-27.804 7 +- 0 16 C18	0.008 +- 0 7 +- 0 10 C14

Train_Data_Set_1
Classifier + Ranker
0 +- 0 1 +- 0 19 C21
0 +- 0 2 +- 0 5 site_domain
0 +- 0 3 +- 0 7 app_domain
0 +- 0 4 +- 0 6 site_category
0 +- 0 5 +- 0 4 banner_pos
0 +- 0 6 +- 0 9 device_model
0 +- 0 7 +- 0 3 C1
0 +- 0 8 +- 0 2 hour
0 +- 0 9 +- 0 8 app_category

Train_Data_Set_2		
CfsSubsetEval + BestFirst	**CfsSubsetEval + Random Search**	**Infogain + Ranker**
10(100 %) 3 site_domain	10(100 %) 3 site_domain	0.063 +- 0.001 1 +- 0 7 device_model
10(100 %) 7 device_model	2(20 %) 5 app_domain	0.055 +- 0.001 2 +- 0 3 site_domain
10(100 %) 9 device_conn_type	10(100 %) 7 device_model	0.032 +- 0.001 3 +- 0 13 C17
9(90 %) 10 C14	8(80 %) 9 device_conn_type	0.03 +- 0.001 4 +- 0 10 C14
10(100 %) 11 C15	2(20 %) 10 C14	0.027 +- 0.001 5 +- 0 17 C21
10(100 %) 12 C16	10(100 %) 11 C15	0.021 +- 0 6 +- 0 15 C19
10(100 %) 14 C18	10(100 %) 12 C16	0.02 +- 0 7 +- 0 14 C18
	8(80 %) 13 C17	0.016 +- 0 8 +- 0 5 app_domain
	10(100 %) 14 C18	

Train_Data_Set_2	
Chisquared +Ranker	**Gainratio + Ranker**
3653.06 +-42.244 1 +- 0 7 device_model	0.036 +- 0.001 1 +- 0 12 C16
3549.994 +-40.329 2 +- 0 3 site_domain	0.024 +- 0.001 2 +- 0 11 C15
2005.541 +-59.209 3.1 +- 0.3 13 C17	0.013 +- 0 3 +- 0 3 site_domain
1889.657 +-88.225 3.9 +- 0.3 10 C14	0.011 +- 0 4 +- 0 14 C18
1510.37 +-33.643 5 +- 0 17 C21	0.011 +- 0 5 +- 0 9 device_conn_type
1185.694 +-21.391 6 +- 0 15 C19	0.009 +- 0 6 +- 0 10 C14
1154.717 +-21.698 7 +- 0 14 C18	0.008 +- 0 7.1 +- 0.3 5 app_domain
	0.008 +- 0 8.2 +- 0.6 13 C17

Train_Data_Set_2
Classifier + Ranker
0 +- 0 1 +- 0 17 C21
0 +- 0 2 +- 0 8 device_type
0 +- 0 3 +- 0 6 app_category
0 +- 0 4 +- 0 5 app_domain
0 +- 0 5 +- 0 4 site_category
0 +- 0 6 +- 0 3 site_domain
0 +- 0 7 +- 0 2 banner_pos
0 +- 0 8 +- 0 7 device_model

Then we performed simple polling with context and selected following attributes.

Selected Atributes
Site_domain
app_domain
C14
C18
C16
C21

Matlab

From the work from the entire dataset that produced the correlations with the click variable and the investigation with the hour variable some conclusion can be drawn.

The hour variable doesn't look to promising in its current form or as a just the hour of the day. The variable C16 looks the most promising with the largest correlation coefficient. Before going any further the raw hour variable is transformed into two new variables, hourofday and dayofweek. My hope is that these variables together will be useful for the prediction.

Also, the device_id and device_ip variables are strings with an enormous amount of unique categories and thus are difficult to deal with. I will ignore these variables in my preliminary feature analysis and reconsider them again later.

The remainder of this section will use the 500k sample exclusively and where all data has been converted to a numerical type for computations.

My first step is to gain an overall understanding of the data by visualizing a scatter plot of each variable against each other and grouped by the state of the click variable, the predictor. The scatter plot shows some interesting features and can help identify variables that may be useful for model. In particular, the figure can also show where a combination of two variables might be suitable predictors. A visual inspections show potential for the following variables: C1, site id, site domain, app domain, device model, C14, C17, C19, C21, and hour of the day and day of the week when combined with other variables in the list.

Next, the mean of each variable was computed for the two groups designated by the click variable. The variables with a percent difference greater than 10% are banner position, app category, device conn type, C16, C20, and C21. The correlation coefficients are computed next for the numerical data.

The largest correlation coefficient with the response variable, click, was 0.13 with C16. Strong correlations greater than 0.9 were detected for two variable pairs. Variables C1 and device type had a correlation coefficient of 0.9 and variables C14 and C17 had a correlation coefficient of 0.98. This suggests redundant information I will plan to only keep one of variables from each pair for modeling later.

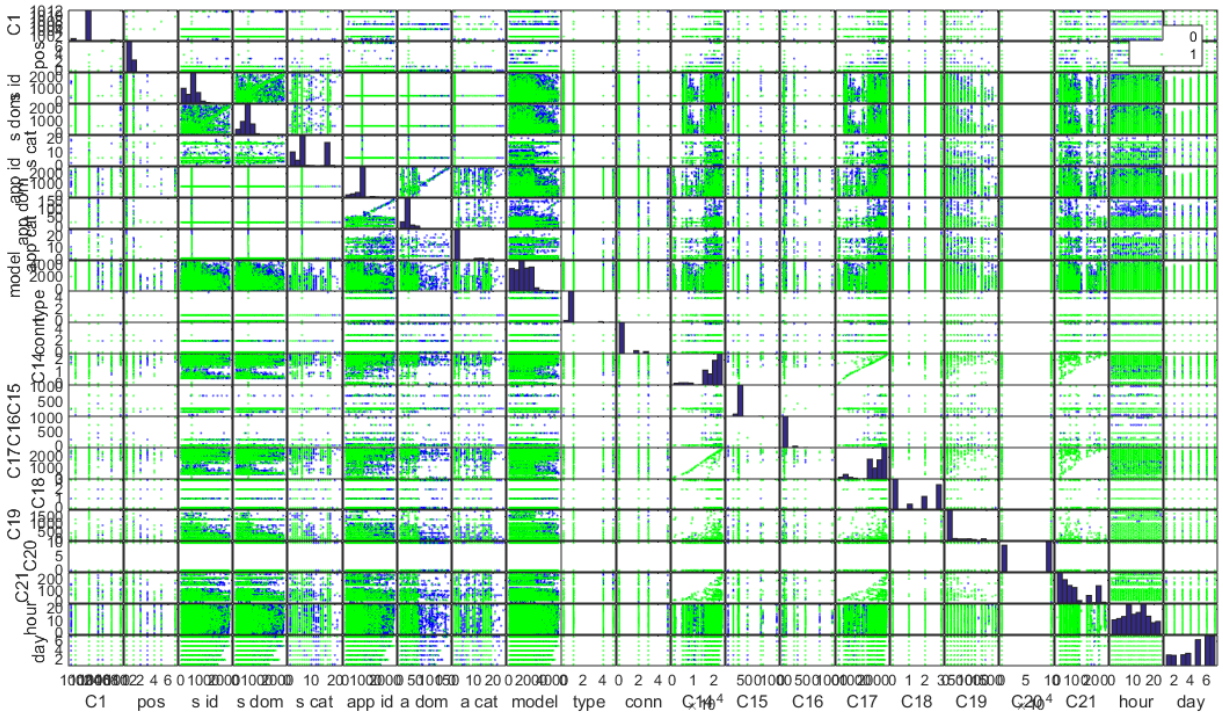


Figure: Scatter plot of 21 variables against each other grouped by the click variable.

Table: Mean values for each variable separated by the click variable state.

Response Variable	Click = 0	Click = 1	% difference
'C1'	1004.98	1004.88	0.01
'ban pos'	0.28	0.32	11.36
's id'	482.82	472.79	2.10
's dom'	536.83	515.38	4.08
's cat'	8.69	8.58	1.29
'app id'	749.90	772.10	2.92
'app dom'	26.87	26.59	1.05
'app cat'	3.09	2.62	16.27
'model'	1230.95	1229.36	0.13
'type'	1.02	0.97	5.31
conn type'	0.36	0.18	69.07
'C14'	18964.18	18236.04	3.91
'C15'	319.22	317.46	0.55
'C16'	57.30	73.16	24.31
'C17'	2125.76	2044.57	3.89
'C18'	1.42	1.49	4.69
'C19'	228.35	225.00	1.48
'C20'	54620.02	46881.36	15.25
'C21'	85.02	73.69	14.28
'hour'	11.28	11.28	0.00
'day'	4.86	4.86	0.00

Next, two advanced methodologies were used to determine feature importance.

- The first methodology was the ReliefF algorithm and provides a rank assignment for each variable. The results for the 500k dataset are shown in the table below. The top five variables identified by the RefliefF algorithm in order are hour of day, C18, app id, app domain, and C15. The use of app id in the model will be difficult since that variable contains 8,552 different categorical values and isn't guaranteed to contain all possible values.
- Next, a sequential feature selection algorithm is used both forwards and backwards for two different base models.
    - The first base model was a simple linear classifier and the forward search selected only two variables, device conn type and C16. The backwards search selected the device model, C14, C16, and C17. The use of the device model variable is subjected to the same difficulties as the app id variable.
    - The next model used for the sequential feature selection was a decision tree model. The forward search identified site id, app id, and C20, while the backwards search identified banner position, site category, app id, app category, C16, C17, C19, and day of the week.
- The last method used to identify features was based on the reduction of out-of-bag error when creating a bagged decision tree or random forest model. This was carried out first by constraining all the variables to be categorical variables and then by only allowing C1, C14, C15, C16, C17, C18, C19, and C21 to be numerical variables. The top five variables identified for the first run with all categorical variables are site id, app id, device model, hour of the day, and day of the week. The top five variables identified for the partial categorical list are device model, C14, C16, hour of the day, and day of the week.

Table: ReliefF feature selection results.

Response Variable	ReliefF Weights	ReliefF Rank
C1	0.0025	20
banner position	0.0124	9
site id	0.0589	12
site domain	0.0569	18
site category	0.0189	21
app id	0.0478	3
app domain	0.0215	4
app category	0.0189	6
device model	0.0984	15
device type	0.0017	17
device conn type	0.0211	7
C14	0.0724	11
C15	0.0039	5
C16	0.0039	8
C17	0.0315	19
C18	0.0043	2
C19	0.0217	16
C20	0.0661	13
C21	0.0127	14
hour of day	0.1059	1
day of week	0.0593	10

The results from all of the feature selection methods are assembled into a summary table presented below. A score of one was given to each variable selected by a certain method and then the scores were tallied across the methods to determine an overall score.

The top scoring variable is C16, which is believed to one of the dimensions in pixels of the ad. Thus this variable could be represented numerically or as a categorical variable. The next highest scoring variables are app id, device model, hour of the day, and day of the week.

The app id and device model variables are problematic for modeling as they have over 8,000 unique categorical values and are not guaranteed to contain all possible values. I believe that these variables could be engineered to a useful state if the true, unobfuscated values were available. I would say the same for the two variables that I have ignored until this point, device id and device ip, which I feel are pseudo-proxies for the identification number. But I could probably extract information from those variables with the real values such as the domain triplet. At this point, I'm going to move forward to modeling without using device id, device ip, app id, and device model.

Instead, I'm going to use the following list of variables to start with and see from the modeling results which variables are needed. The variable I'm starting with are banner position, site domain, site category (site domain and site category will stand in for site id), app domain, app category (app domain and app category will stand in for app id), device conn type, C14, C16, hour of the day, and day of the week.

Table: Correlation coefficients for the 500k dataset.

	click	C1	Ban pos	site id	site dom	site cat	app id	app dom	app cat	device model	device type	device conn type	C14	C15	C16	C17	C18	C19	C20	C21	hour of day	day of week
click	1.00	-0.04	0.03	-0.01	-0.04	-0.01	0.03	-0.01	-0.04	0.00	-0.04	-0.08	-0.06	-0.03	0.13	-0.05	0.02	0.00	-0.06	-0.06	0.00	0.00
C1	-0.04	1.00	0.29	-0.05	0.02	0.03	-0.09	0.00	0.09	0.06	0.90	0.19	0.06	0.12	0.06	0.07	-0.04	0.00	-0.04	0.04	0.01	0.02
banner position	0.03	0.29	1.00	0.28	-0.36	0.53	0.14	0.03	-0.22	0.05	0.32	-0.08	-0.01	0.06	0.03	-0.03	0.10	0.13	0.05	-0.10	0.00	0.00
site id	-0.01	-0.05	0.28	1.00	-0.02	0.42	-0.01	0.00	0.02	0.03	-0.05	0.00	-0.01	0.00	-0.03	-0.01	0.29	0.08	0.09	-0.15	0.02	0.03
site domain	-0.04	0.02	-0.36	-0.02	1.00	-0.50	-0.10	-0.03	0.15	-0.01	0.02	0.10	0.00	0.06	-0.11	0.01	-0.14	-0.03	-0.06	0.09	0.01	-0.02
site category	-0.01	0.03	0.53	0.42	-0.50	1.00	0.14	0.05	-0.21	0.02	-0.01	-0.14	0.00	0.00	-0.11	-0.01	0.15	0.11	0.11	-0.12	-0.01	0.03
app id	0.03	-0.09	0.14	-0.01	-0.10	0.14	1.00	0.05	-0.36	-0.01	-0.10	-0.23	-0.05	0.04	0.07	-0.05	-0.02	0.08	-0.03	0.03	-0.05	-0.03
app domain	-0.01	0.00	0.03	0.00	-0.03	0.05	0.05	1.00	-0.18	0.00	0.00	-0.02	0.02	0.03	0.01	0.02	-0.02	0.01	0.18	-0.09	0.04	-0.05
app category	-0.04	0.09	-0.22	0.02	0.15	-0.21	-0.36	-0.18	1.00	0.00	0.08	0.23	0.00	0.04	-0.06	0.02	0.08	0.02	-0.06	0.06	0.03	0.06
device model	0.00	0.06	0.05	0.03	-0.01	0.02	-0.01	0.00	0.00	1.00	0.06	0.00	-0.01	-0.01	0.00	-0.01	0.06	0.01	0.01	-0.03	0.00	0.00
device type	-0.04	0.90	0.32	-0.05	0.02	-0.01	-0.10	0.00	0.08	0.06	1.00	0.21	0.05	0.18	0.07	0.05	-0.05	0.00	-0.05	0.04	0.01	0.01
device conn type	-0.08	0.19	-0.08	0.00	0.10	-0.14	-0.23	-0.02	0.23	0.00	0.21	1.00	0.07	0.07	-0.01	0.08	-0.06	-0.01	0.09	0.06	0.03	-0.05
C14	-0.06	0.06	-0.01	-0.01	0.00	0.00	-0.05	0.02	0.00	-0.01	0.05	0.07	1.00	0.00	0.04	0.98	-0.23	-0.13	0.02	0.41	-0.05	0.13
C15	-0.03	0.12	0.06	0.00	0.06	0.00	0.04	0.03	0.04	-0.01	0.18	0.07	0.00	1.00	-0.07	0.00	0.01	0.05	0.01	0.00	0.00	-0.01
C16	0.13	0.06	0.03	-0.03	-0.11	-0.11	0.07	0.01	-0.06	0.00	0.07	-0.01	0.04	-0.07	1.00	0.05	0.08	-0.07	-0.05	-0.08	0.01	-0.02
C17	-0.05	0.07	-0.03	-0.01	0.01	-0.01	-0.05	0.02	0.02	-0.01	0.05	0.08	0.98	0.00	0.05	1.00	-0.25	-0.13	0.01	0.42	-0.05	0.15
C18	0.02	-0.04	0.10	0.29	-0.14	0.15	-0.02	-0.02	0.08	0.06	-0.05	-0.06	-0.23	0.01	0.08	-0.25	1.00	0.09	0.01	-0.54	0.02	-0.08
C19	0.00	0.00	0.13	0.08	-0.03	0.11	0.08	0.01	0.02	0.01	0.00	-0.01	-0.13	0.05	-0.07	-0.13	0.09	1.00	0.09	-0.14	0.00	-0.05
C20	-0.06	-0.04	0.05	0.09	-0.06	0.11	-0.03	0.18	-0.06	0.01	-0.05	0.09	0.02	0.01	-0.05	0.01	0.01	0.09	1.00	-0.04	0.02	-0.04
C21	-0.06	0.04	-0.10	-0.15	0.09	-0.12	0.03	-0.09	0.06	-0.03	0.04	0.06	0.41	0.00	-0.08	0.42	-0.54	-0.14	-0.04	1.00	-0.07	0.20
hour of day	0.00	0.01	0.00	0.02	0.01	-0.01	-0.05	0.04	0.03	0.00	0.01	0.03	-0.05	0.00	0.01	-0.05	0.02	0.00	0.02	-0.07	1.00	-0.02
day of week	0.00	0.02	0.00	0.03	-0.02	0.03	-0.03	-0.05	0.06	0.00	0.01	-0.05	0.13	-0.01	-0.02	0.15	-0.08	-0.05	-0.04	0.20	-0.02	1.00

Response Variable	Plot Matrix	% Diff. of Mean	Correlation Coefficients	ReliefF	Sequential-linear-forward	Sequential-linear-backward	Sequential-tree-forward	Sequential-tree-backward	Random Forest-partial cat	Random Forest-all cat	Total Score
C1	1		a								**1**
banner position		1						1			**2**
site id	1						1			1	**3**
site domain	1										**1**
site category								1			**1**
app id				1			1	1		1	**4**
app domain	1			1							**2**
app category		1						1			**2**
device model	1					1			1	1	**4**
device type			a								**0**
dev conn type		1			1						**2**
C14	1		b			1			1		**3**
C15				1							**1**
C16		1	1		1	1		1	1		**6**
C17	1		b			1		1			**3**
C18				1							**1**
C19	1							1			**2**
C20		1					1				**2**
C21	1	1									**2**
hour of day	1			1					1	1	**4**
day of week	1							1	1	1	**4**

Initial Modeling & Baseline

Coin-Flip Baseline

All 0.5 Benchmark, score = 0.6931472

The public baseline set by Kaggle. Random guessing with all predictions set to a probability of 0.5 of clicking the ad

Random Selection Baseline

Random 17% click, score = 9.5526002

The public baseline set by Kaggle randomly selects 17% of the rows to click. Matching population distribution in sample data, 17% of the test data was randomly selected with a uniform probability and given a 100% probably of clicking the ad, the remaining test data was given a 0% probability of clicking the ad. So, unless you get really lucky and pick the exact right entries you're going to fail.

Single Model Baseline – Random Forest

Our first submission was a Random Forest using Azure ML using all of the columns with no feature engineering and with a stratified sample size of 10%. This achieved a score of 0.4289.

Comparison of Multiple Classifiers & Settings

Azure ML - 7-Variable Boosted Decision Tree

Similar to the feature selection method discussed earlier, multiple classifiers were evaluated iteratively to balance F-score and AUC against the complexity of the model.

Classifier	*F-score*	*AUC*
SVM	0.005	0.545
Neural Net	0.165	0.631
Decision Jungle	0.072	0.683
Logistic Regression	0.004	0.609
Boosted D-Tree*	**0.126**	**0.689**

The Boosted Decision Tree classifier was determined to best balance complexity against predictive power. Finally the number of leaves per tree was tested using the same approach to experiment with a slightly more tightly fit tree, resulting in the following statistics:

Number of Leaves	*F-score*	*AUC*
20	0.126	0.689
100*	**0.145**	**0.683**

The 7-variable Boosted Decision Tree with 100 trees was determined to improve F-Score while maintaining AUC, resulting in the following ROC chart:

Kaggle Submissions & Results

So far, our best model is performing almost twice as good as randomly guessing, which is encouraging. The boosted decision tree appears to perform better than the SVM but the SVM should be given a chance with optimized parameters.

Table: Summary table of results for submitted models. (* used in ensemble model)

Name	Description	Feature Engineering	Score
Random guess benchmark	Random guessing with all predictions set to a probability of 0.5 of clicking the ad	None	0.6931472
Boosted Tree-10% data	Boosted decision tree from Azure ML with a maximum of 20 leaves per a tree, minimum of 10 samples per leaf, a learning rate of 0.2, the number of trees set to 100, and trained at the entire dataset.	None	0.3993284
SVM	SVM from Azure ML with only 10% of training data.	None	0.4352991
Boosted Tree-all data Model A*	Boosted decision tree from Azure ML with a maximum of 20 leaves per a tree, minimum of 10 samples per leaf, a learning rate of 0.2, the number of trees set to 100, and trained on the entire dataset.	None	0.3990766
Boosted Tree-all data Model b	Boosted decision tree from Azure ML with a maximum of 40 leaves per a tree, minimum of 100,000 samples per leaf, a learning rate of 0.2, the number of trees set to 100, and trained on the entire dataset.	None	0.4049064
Random 17% Benchmark	Matching population distribution in sample data, 17% of the test data was randomly selected with a uniform probability and given a 100% probably of clicking the ad, the remaining test data was given a 0% probability of clicking the ad.	None	9.5526002
Baseline Neural Network	Azure ML default parameters neural network with entire Kaggle dataset	None	0.4678139
Logistic Regression	Linear logistic regression from R, single predictor banner_pos, 500k sample set	Categorical	0.4411265
Logistic Regression*	Linear logistic regression from R, C14 and C16 with interactions only, 500k sample set	None	0.4361704
Logistic Regression	Linear logistic regression from R, C14 only, 500k sample set	None	0.4413473
Boosted Tree-Scarpa*	Azure ML, subset of data		5.5590110
Ensemble Model	Simple linear average of Boosted Tree-all data Model A, Logistic Regression with C14 and C16, and Boosted Tree-Scarpa	Ensemble	0.4557994
Ensemble Model	Simple linear average of Boosted Tree-all data Model A, and Logistic Regression with C14 and C16,	Ensemble	0.4487923

Feature Engineering & Transformations

Supervised Discretization

Categorical features that have large numbers of nominal values present problems for ML algorithms since most (if not all) need to first create Boolean dummy variables through binarization. This can lead to very sparse matrices which are difficult to generalize from since each dummy variable needs an appropriate sample. One way to rectify this is through collapsing the nominal values into a much smaller number of bins. There are different approaches to do this, but one that was incorporated into our model was the supervised approach, whereby each nominal value was tagged with the percentage of "clicks" associated with that value. Then each nominal value was ranked in terms of this percentage and a bin was assigned based on its quantile.

Of course when this is done information is lost regarding the relationships between individual nominal values. In the future we will extend this process to identify the set of features most relevant when coupled together and performing the same discretization process on them.

Ensemble Modeling Results

Early in the project we decided we would attempt to ensemble models that were created by individual team members. The following is a description and evaluation of that approach. The ensemble was composed of three models. Two of the models were boosted decision trees from Azure ML and the third was a logistic regression model in R using only C14 and C16 as predictors. The Kaggle scores for the individual models were 0.3990766, 5.5590110, and 0.4361704, respectively. The ensemble model was formed by taking the simple linear averages of the estimated probabilities from each of the three individual models. The Kaggle score of the ensemble model was 0.4557994, which isn't as good as either our best scoring boosted decision tree model or the logistic regression, but the ensemble is scoring much better than the other boosted decision tree model. Overall, the ensembling isn't improving our overall score but more work is needed to know for sure.

Final Submission Results

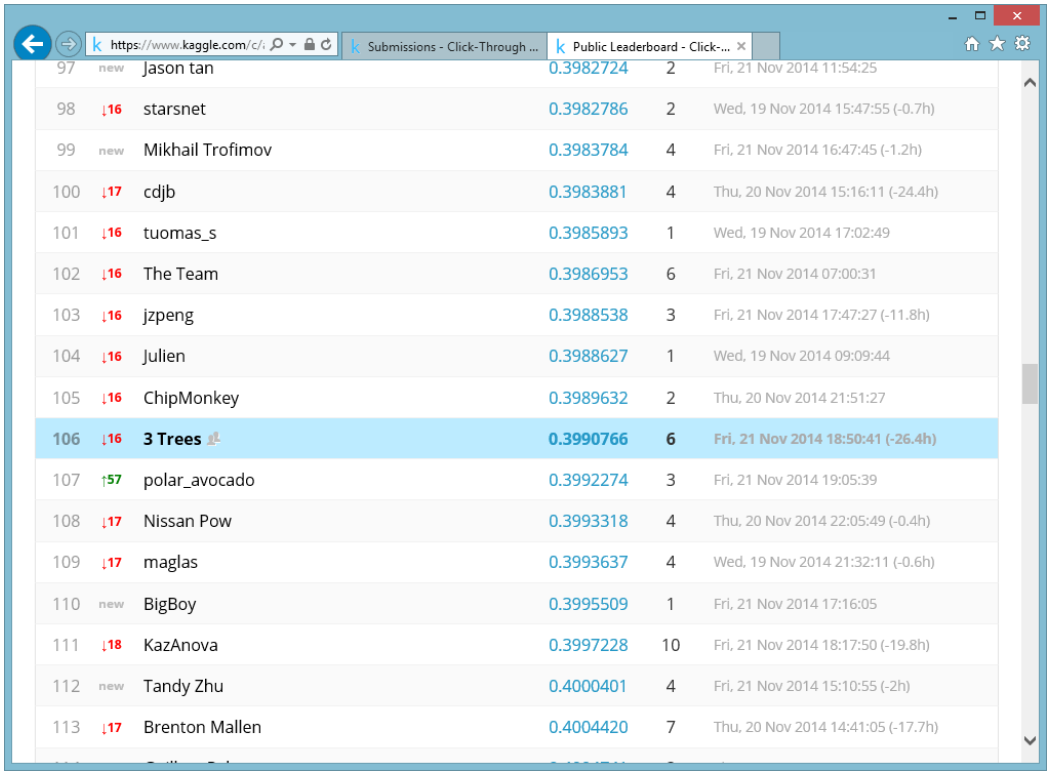
Ultimately the following model was submitted to Kaggle as the team's final submission.

Azure ML

Team baseline, score = 0.3990766

Best score as of November 21, 2014 11 am.

Boosted decision tree from Azure ML with a maximum of 20 leaves per a tree, minimum of 10 samples per leaf, a learning rate of 0.2, the number of trees set to 100, and trained at the entire dataset.



97	new	Jason tan	0.3982724	2	Fri, 21 Nov 2014 11:54:25
98	↓16	starsnet	0.3982786	2	Wed, 19 Nov 2014 15:47:55 (-0.7h)
99	new	Mikhail Trofimov	0.3983784	4	Fri, 21 Nov 2014 16:47:45 (-1.2h)
100	↓17	cdjb	0.3983881	4	Thu, 20 Nov 2014 15:16:11 (-24.4h)
101	↓16	tuomas_s	0.3985893	1	Wed, 19 Nov 2014 17:02:49
102	↓16	The Team	0.3986953	6	Fri, 21 Nov 2014 07:00:31
103	↓16	jzpeng	0.3988538	3	Fri, 21 Nov 2014 17:47:27 (-11.8h)
104	↓16	Julien	0.3988627	1	Wed, 19 Nov 2014 09:09:44
105	↓16	ChipMonkey	0.3989632	2	Thu, 20 Nov 2014 21:51:27
106	**↓16**	**3 Trees**	**0.3990766**	**6**	**Fri, 21 Nov 2014 18:50:41 (-26.4h)**
107	↑57	polar_avocado	0.3992274	3	Fri, 21 Nov 2014 19:05:39
108	↓17	Nissan Pow	0.3993318	4	Thu, 20 Nov 2014 22:05:49 (-0.4h)
109	↓17	maglas	0.3993637	4	Wed, 19 Nov 2014 21:32:11 (-0.6h)
110	new	BigBoy	0.3995509	1	Fri, 21 Nov 2014 17:16:05
111	↓18	KazAnova	0.3997228	10	Fri, 21 Nov 2014 18:17:50 (-19.8h)
112	new	Tandy Zhu	0.4000401	4	Fri, 21 Nov 2014 15:10:55 (-2h)
113	↓17	Brenton Mallen	0.4004420	7	Thu, 20 Nov 2014 14:41:05 (-17.7h)

A sample of the Kaggle public leaderboard for Avazu Click Through Rate Challenge as of November 21, 2014 11:15 am.