MOBILE ADVERTISING PREDICTION & CLASSIFICATION

By

Nengi Harry

INTRODUCTION

This project aims to predict which mobile application platform an advertisement should be placed on while minimizing total expected cost to the consumers. The premise is this; an application developer has an app that he would like to advertise so that consumers can install it. To ensure visibility the developer can go to a mobile advertisement platform that will then publish the app on several other apps (publishers) on its platform, and when users see it, they can either install or not-install, below is a flow chart of this process.

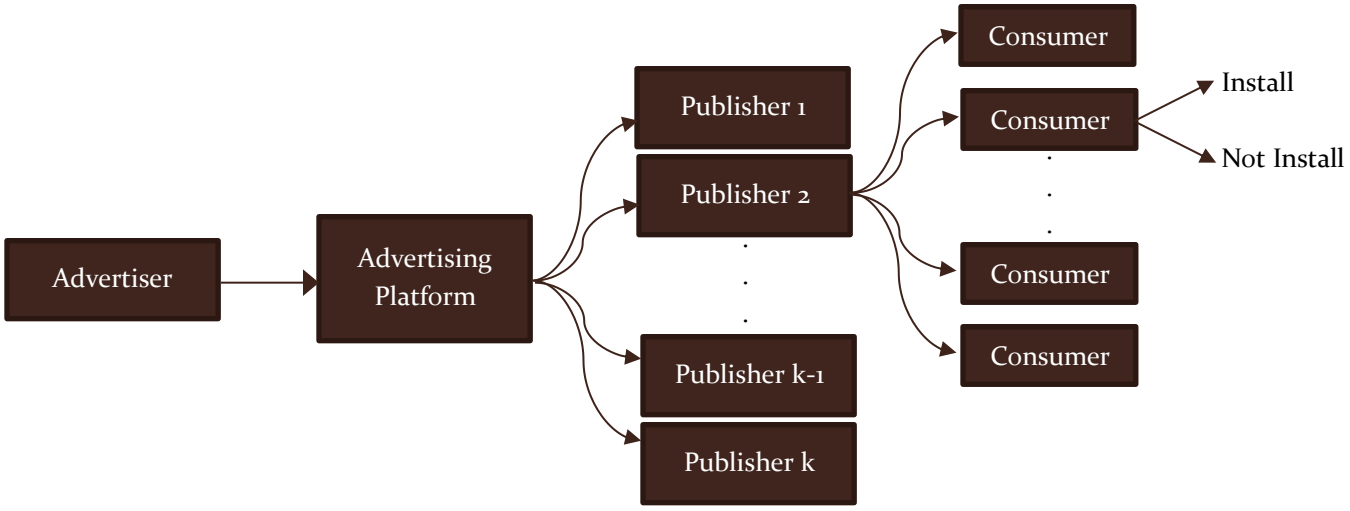


Image was copied from original problem statement

The problem type is a binary classification problem and it was resolved using logistic regression in SAS. Also, the dataset is imbalanced, success rate – number of installations make up 0.81% of all site visits by consumers. Random Oversampling proved to be the most effective improving the AUC score to 70%.

The model to be built is generic to publishers, as it was built to be applied to any set of publishers that have the same customer demographics as shown in the data set.

EXPLORATORY ANALYSIS

The aim of the exploratory analysis was to understand the features in the dataset the impact they have on consumer outcome, and to utilize observed relationships to perform feature engineering to improve prediction outcome

First off, an overview of the dataset is provided so each of the variables and function in the dataset is understood. Next each variable, combination of variables and newly engineered features was analyzed and tested to understand their importance to the outcome. A Chi-Square test of independence was performed with an alpha level of 0.05 - if the p-value was less than 0.05 then its significant and the variables can be said to have a dependent relationship, if it is greater than the relationship is not significant and the variables are independent.

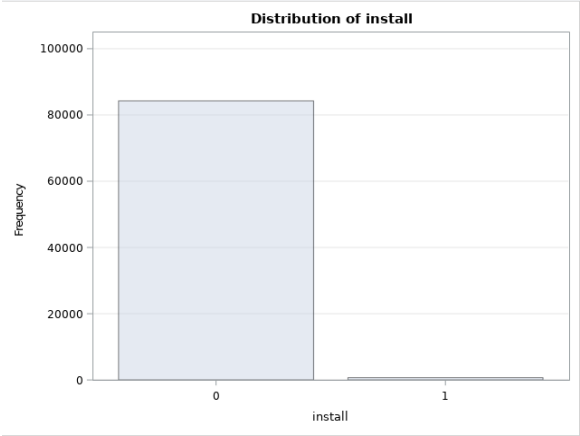
Data Overview: Below is a table providing a description of all the variables in the dataset

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
5	device_height	Num	8	BEST12.	BEST32.	Display Height (in pixels)
10	device_make	Char	8	$8.	$8.	Device Manufacturer
8	device_os	Char	6	$6.	$6.	Phone OS Version
7	device_platform	Char	7	$7.	$7.	Phone OS Type (Andriod/IOS)
2	device_volume	Num	8	BEST12.	BEST32.	Device Volume when Ad is displayed
6	device_width	Num	8	BEST12.	BEST32.	Display Width (in pixels)
1	install	Num	8	BEST12.	BEST32.	Customer Installed App (Yes = 1, No = 0)
9	language	Char	5	$5.	$5.	Language settings on device
11	publisher_id	Num	8	BEST12.	BEST32.	Publisher Id
4	resolution	Num	8	BEST12.	BEST32.	Display Resolution (pixels per inch)
3	wifi	Num	8	BEST12.	BEST32.	Wifi Enabled (Yes = 1, No = 0)

Proportion of Installations: As stated, the number of app installations in the training dataset is extremely low at 0.81%, making the dataset imbalanced with a disproportionately large number of consumers not installing.

install	Frequency	Percent
0	84246	99.19
1	692	0.81

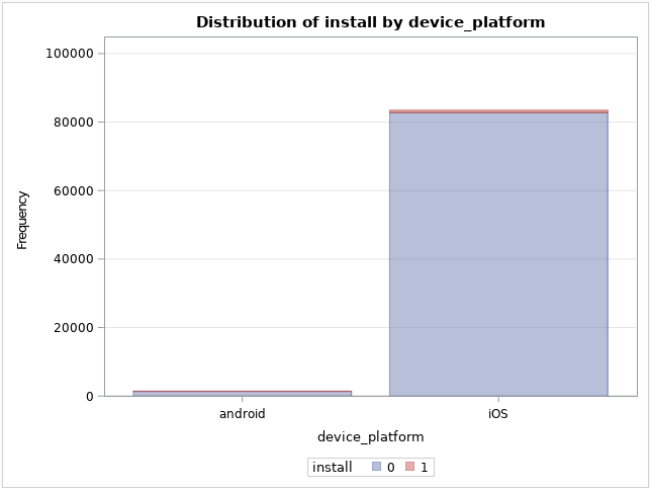


Number of Publishers: The dataset had 1304 publishers, only 57 or 0.044% had success of consumers installing the app as seen below. The chi-square test indicates there is a dependent relationship between publisher's and wether a consumer installs. This means that are certain types of apps/publishers on which the advertiser can rely on. A way to measure this would be to calculate a success rate metric which would be percentage of installations a publisher has had based on the number of customer visits.

install	Frequency
0	1247
1	57

Proportion of Operating Systems Type Recorded: The most operating system used was the Android. It accounts for 0.8% of all installations which is about 99% of the install group. A chi-square test indicates that the relationship between the operating system and install is independent. Which means the phone operating system most likely has no influence on a consumer's decision to install.

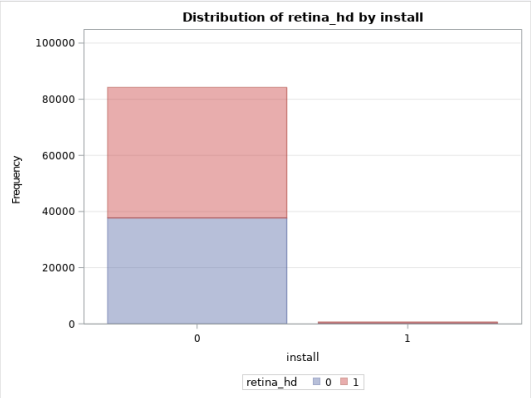
Table of install by device_platform			
	device_platform		
install	android	iOS	Total
0	1422 1.67	82824 97.51	84246 99.19
1	12 0.01	680 0.80	692 0.81
Total	1434 1.69	83504 98.31	84938 100.00



Proportion of installations by device make: From the above chart in the data set most of the consumers used android phones. The charts below show the proportions of installations from iPhones, iPads and Apple HD devices.

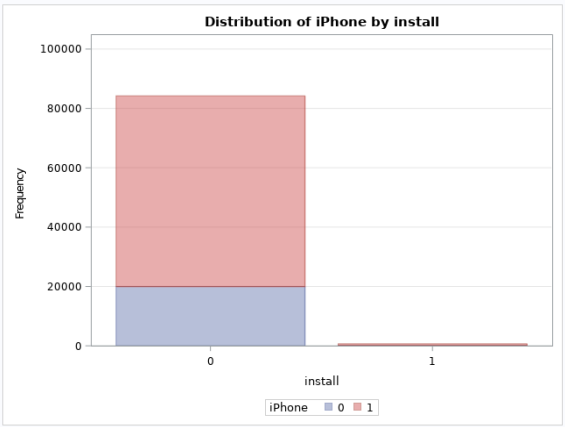
1. **By Apple HD Device**

Table of retina_hd by install			
	install		
retina_hd	0	1	Total
0	37758 44.45	308 0.36	38066 44.82
1	46488 54.73	384 0.45	46872 55.18
Total	84246 99.19	692 0.81	84938 100.00



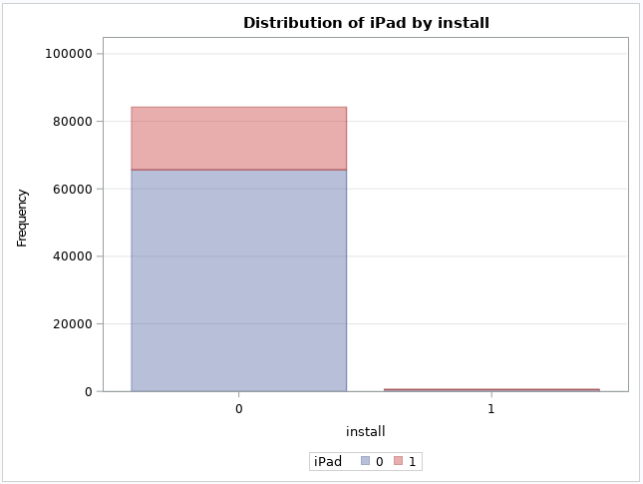
2. **By iPhones**

Table of iPhone by install			
	install		
iPhone	0	1	Total
0	20022 23.57	184 0.22	20206 23.79
1	64224 75.61	508 0.60	64732 76.21
Total	84246 99.19	692 0.81	84938 100.00



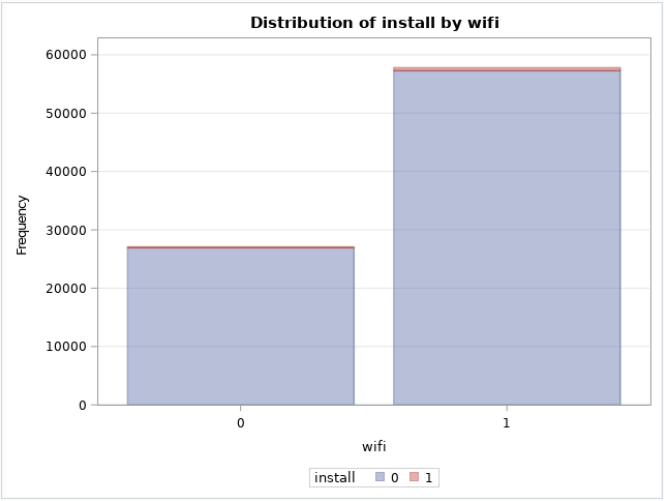
3. **By iPads**

Table of iPad by install			
	install		
iPad	0	1	Total
0	65683 77.33	520 0.61	66203 77.94
1	18563 21.85	172 0.20	18735 22.06
Total	84246 99.19	692 0.81	84938 100.00



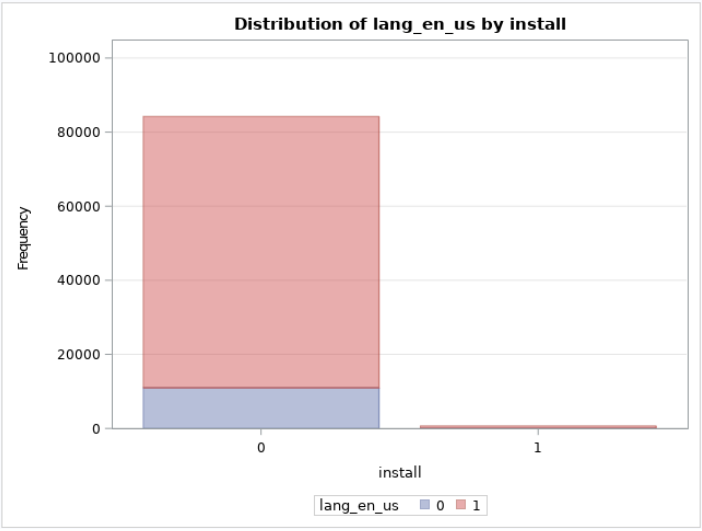
WIFI Use: 0.61% of the users which is about 75% of the installation group had WIFI enabled when they installed the app. The chi-square test was significant and so the relationship is most likely dependent. It can be inferred that users might be more likely to install the app when on wifi

Table of install by wifi			
	wifi		
install	0	1	Total
0	31.72	67.47	99.19
1	0.20	0.61	0.81
Total	27113 31.92	57825 68.08	84938 100.00



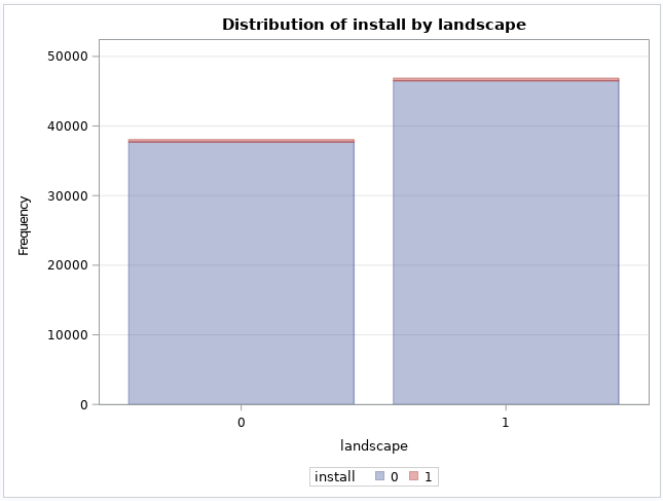
Language Settings: There are about 211 languages and a chi-square test between them, and installation is independent. Instead the languages are divided into 3 groups – "US English", "English" and "Others". The chis-square test was only significant for US English, 0.73% of them which is about 90% of the installation group had their phone language settings on US English.

Table of lang_en_us by install			
	install		
lang_en_us	0	1	Total
0	11013 12.97	71 0.08	11084 13.05
1	73233 86.22	621 0.73	73854 86.95
Total	84246 99.19	692 0.81	84938 100.00

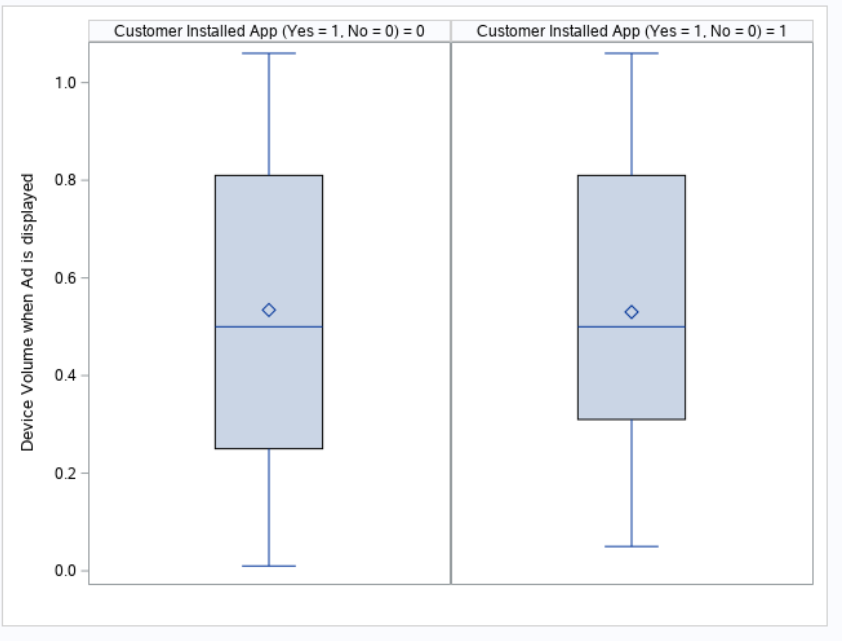


Screen Orientation: A screen orientation variable called portrait was created to capture the position of the screen when the advert was viewed; when the value is 1 screen orientation was portrait at 0 it was landscape. 0.43% of the users which comes down to 53% in the install group had the phone in landscape mode while 0.39% of users which is 47% in the install group had the phone in portrait mode. There is little to no variation with when customers install, and this is confirmed by the chi-square test that indicates the variables are independent.

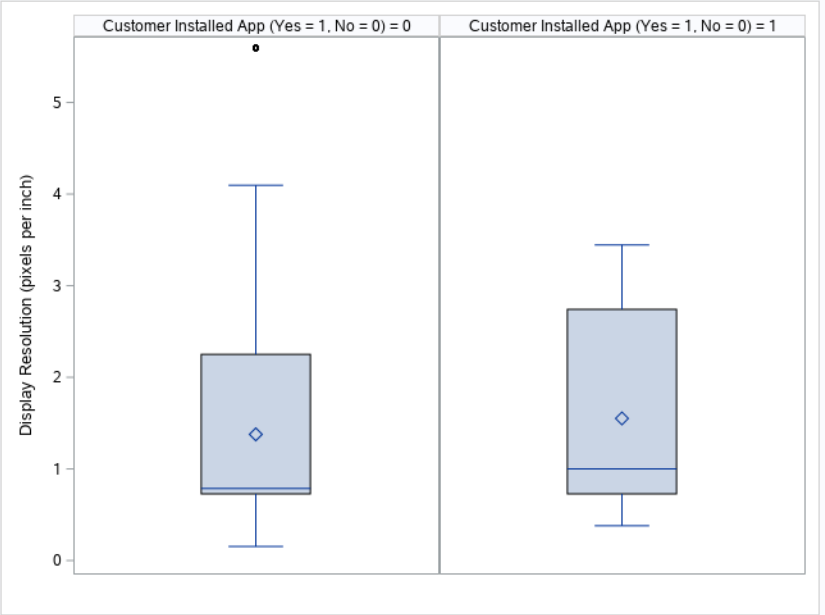
Table of install by landscape			
	landscape		
install	0	1	Total
0	37724 44.41	46522 54.77	84246 99.19
1	330 0.39	362 0.43	692 0.81
Total	38054 44.80	46884 55.20	84938 100.00



Device Volume: The device volume appears to be within the same range for both people who installed, and consumers that did not install, which could mean that volume does not influence installation. A chi-square test also confirms this because the result is insignificant meaning both variables are independent.



Resolution: The resolution range for consumer's who installed appears to be smaller than for consumers who did not install, which could mean there might be some slight relationship. The various chi-square tests are inconclusive, but the models will help determine if there is a true effect or not.



DATA PREPARATION & FEATURE ENGINEERING

The dataset came with two datasets for train and test. For the model building and selection process, the training set was utilized with cross validation. The test set was used for the final classification.

To prepare for modelling, the categorical variables where dummy coded, and new variables (highlighted in **bold**) where created by calculating new statistics to measure characteristics. Redundant variables or variables that could be inferred from other variables were dropped:

- The **portrait** variable was created by comparing the device_height and device_width variables. I dropped both variables because they can be determined from the resolution and portrait variable.

- **Publisher_install_rate** is the installation success rate for each publisher. It is the sum of installs a publisher has divided by the number of visits for that publisher. I dropped the publisher id because the install rate is constant for each publisher which can then be used to identify types of publishers and most importantly can be extended to new datasets because all that will be required is their past success rate.

- An indicator variable for device_platform was created – for the various iOS levels from **iOS10 to iOS 7**.

- Based on Apple's marketing brand names and the percentage of installations the devices that fall under the class of Apple's retinaHD where grouped under a variable **retina_HD**, iPhones and iPads were put in the variabes **iPhone** and **iPad** respectively

- Also, indicator variables where created for the language settings based on percentage of successful installs. The US English setting **lang_en_us** was significant in a chi square test.

MODEL BUILDING & SELECTION

Resolve Imbalance: SMOTE

Prior to building the model imbalance in the train dataset was rectified by using Synthetic Minority Oversampling Technique (SMOTE). SMOTE is a method of oversampling the minority class by creating new synthetic samples from the minority class. Below is a general guideline for the SMOTE algorithm.

- For the k nearest neighbors of a given observation
- Compute the difference between the sample and a given nearest neighbor
- Multiply the difference by a random number between 0 and 1. The final sample is a random point on the line between the 2 observations.

As shown above the SMOTE algorithm is only applicable to continuous features, as such the Synthetic Minority Oversampling Technique-Nominal Continuous (SMOTENC) method from imbalance library in Python was used. The SMOTENC method extends to handle both continuous and categorical variables. The SMOTENC process is as follows

- For continuous variables utilize the SMOTE process described above
- For categorical variables,
    - Calculate the median of the standard deviation of all the continuous variables in the minority class.
    - Calculate the Euclidean distance between the feature vectors for which k-nearest neighbors have been identified.
        - If the categorical data between 2 vectors differ, include the previously calculated median in the Euclidean distance calculation
    - The majority value amongst the k-nearest neighbors is then assigned to the new observation.

The picture below explains the computation process.

Table 6: Example of nearest neighbor computation for SMOTE-NC.

F1 = 1 2 3 A B C [Let this be the sample for which we are computing nearest neighbors]
F2 = 4 6 5 A D E
F3 = 3 5 6 A B K
So, Euclidean Distance between F2 and F1 would be:
Eucl = sqrt[$(4-1)^2 + (6-2)^2 + (5-3)^2 + Med^2 + Med^2$] Med is the median of the standard deviations of continuous features of the minority class.
The median term is included twice for feature numbers 5: B→D and 6: C→E, which differ for the two feature vectors: F1 and F2.

Model Development: Logistic Regression

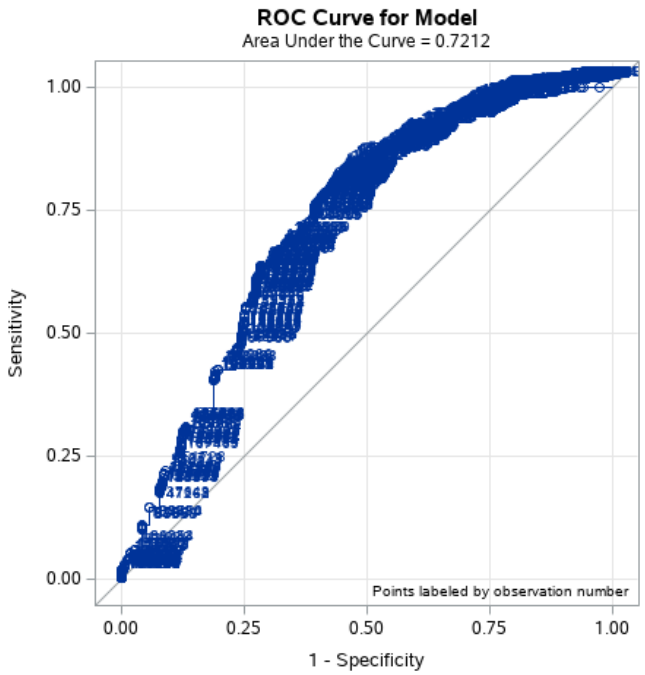
Logistic regression is used to classify data with categorical outcomes of which the simplest case is the binary outcome that can be encoded as 0 or 1. This is achieved by applying a sigmoid function to constrain the output of a linear probability model to 0 or 1. This output is then the probability that a given outcome is either observed (1) or not observed (0). The set of equations below

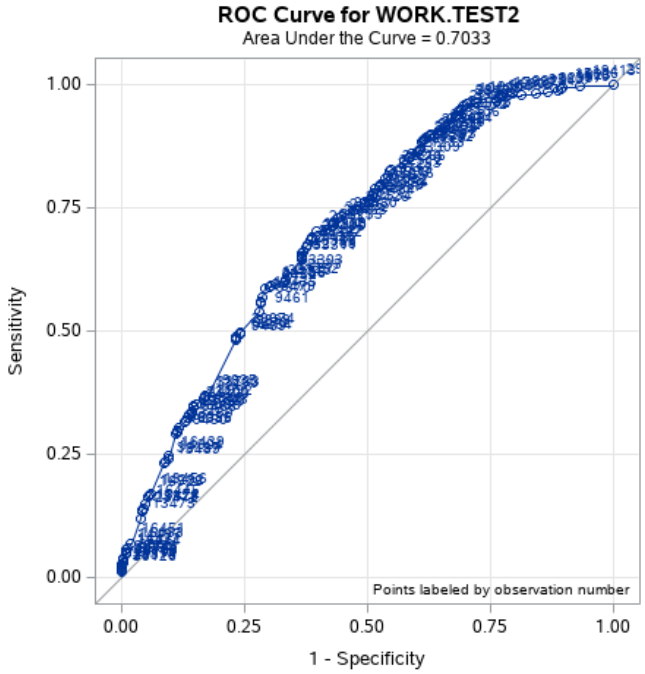
The model was built on the smote dataset using stepwise selection to choose features for the model and the model with the largest AUC was chosen. Below is the model's parameter estimates and ROC on train and test data. Final values are 72.12% for train data and 70.33% for test data

Parameter Estimates

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-2.6021	0.0317	6717.3809	<.0001
resolution	1	-0.0278	0.0129	4.6551	0.0310
publisher_install_ra	1	117.7	2.4214	2361.9296	<.0001
lang_en_us	1	0.8122	0.0208	1521.0657	<.0001
landscape	1	-0.0847	0.0269	9.9321	0.0016
wifi	1	0.5762	0.0126	2076.7898	<.0001
install_rate_sq	1	-175.5	3.6088	2364.0106	<.0001
resolutio*publisher_	1	21.7285	1.2735	291.1165	<.0001
publisher_*landscape	1	29.2590	2.8732	103.7003	<.0001
resolu*publis*landsc	1	-13.2662	1.0178	169.8846	<.0001

Roc Curve for Train and Test





Data Classification

To classify the data and determine when an advert should be displayed, a probability that minimizes total cost to a consumer must be used as the classification threshold, if the probability for a consumer is above the threshold, the advert is shown, when its less it is not shown.

The total expected cost to a consumer is determined as follows:

- Showing an ad to a consumer who would not install results in an inconvenience cost 'C1'. This is a False Positive 'FP'.
- Not showing the ad to a consumer who would install results in a missed opportunity cost 'C2'. This is a False Negative 'FN'.
- The cost ratio of C2 to C1 can be any of the following: 25, 50, 100, 200

Following the definition above the formula below was used to determine what the total cost should be as follows:

$$TC = C_1 * FP + C_2 * FN$$

where TC is Total Cost.

Let cost ratio be defined as R, where $R = C_2/C_1$, then $C_2 = C_1 * R$

Therefore $TC = C_1*FP + C_1*R*FN$

Based on the formula above the best model with a minimal false negative rate has the best effect of minimizing the total cost. As such the model best Sensitivity (maximum) and minimal False Negative rate will be chosen

To classify the data the process below was utilized.

1. Sort the probabilities in the classification table by ascending order
2. Calculate the total cost for cost ratios 25, 50, 100 and 200 for each of the probabilities using the formula. Assume C1 is $1.
3. Choose the probability associated with the minimum cost and minimum false negative.

The table below is a summary of the probabilities at which data should be classified given for a given cost ratio and a confusion matrix for classification at threshold of 0.7

Proabilities and Minimum Costs At 25, 50, 100, 200

Obs	probability	mincost
1	0.06	75.00
2	0.7	95.05
3	0.7	122.10
4	0.8	173.30

The FREQ Procedure

Percent

Table of install by show_ad

install	show_ad: no	yes	Total
0	86.07	13.07	99.13
1	0.59	0.28	0.87
Total	31544 86.66	4857 13.34	36401 100.00