Modelling App Installations and Minimizing Total Costs

Introduction

A mobile app creator advertises his/her app on other apps through a mobile advertising platform to increase the app installs. The advertisement is shown across 10 different apps on both iOS and Android devices with different mobile device characteristics. The objective of the report is to develop a probability model that deems profitable to the company in terms of identifying right customers who would install the app. This report tries to develop the best model that minimizes the net expected advertising cost incurred to the app developer.

About Data

Each observation corresponds to one ad shown to a consumer on a particular publisher app. The observation contains information about the publisher id, consumer's device characteristics, and whether the advertiser's app was installed or not. There are such 121, 339 instances recorded in the dataset. Following is the description of the variables:

Variable	**Type**	**Description**
publisher_id_class	Categorical	Publisher Id (10 Classes)
device_make_class	Categorical	Device Manufacturer (10 Classes)
device_platform_class	Categorical	Phone OS Type (iPhone / Android)
device_os_class	Categorical	Phone OS Version (10 Classes)
device_height	Numerical	Display Height (in pixels)
device_width	Numerical	Display Width (in pixels)
resolution	Numerical	Display Resolution (pixels per inch)
device_volume	Numerical	Device Volume when Ad was displayed
Wifi	Binary	Whether WiFi was enabled when ad was displayed (Yes = 1, No = 0)
install	Binary	Whether Consumer Installed Advertiser's App (Yes = 1, No = 0)

EDA

Response Variable:

A quick look into the data reveals that less than 1% of the customer base have installed the app. Therefore, the dataset is highly imbalanced. Modeling such rare outcomes might lead to biased maximum likelihood estimates. Hence models are built on both original dataset as well as oversampled dataset which are discussed in later sections.

Frequency Counts of Orignal Dataset

The FREQ Procedure

install	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	120331	99.17	120331	99.17
1	1008	0.83	121339	100.00

Categorical Variables:

The frequency table containing install counts of various combinations of device manufacturers, platforms and OS classes shows that most of the data comes from iOS users. This might be attributed to the fact that advertiser of the app prefers to advertise on iOS devices. The only Android device that was recorded in the dataset is of class 10 manufacturer. And the app was advertised only on class 10 publisher app for such android users.

Table: Platform, Make & OS Version vs Install

			install	
			0	1
			N	N
device_platform_class	device_make_class	device_os_class		
android	10	10	2049	16
iOS	1	1	18329	143
		2	2067	20
		3	3969	33
		4	3485	38
		5	1276	10
		6	1376	10
		8	796	1
		9	707	.
		10	1860	16
	2	1	12825	108
		2	1933	8
		3	3250	12
		4	2331	20
		5	884	5
		6	912	11
		8	627	5
		9	393	3
		10	2093	8
	3	1	5992	44
		2	1054	4
		3	1359	7
		4	1036	10
		5	605	10
		6	279	2
		7	54	.
		8	367	5
		9	335	1
		10	1264	2
	4	1	3950	51
		2	1255	6
		3	538	3
		4	992	16
		5	713	4

Table: Platform, Make & OS Version vs Install

			install	
			0	1
			N	N
device_platform_class	device_make_class	device_os_class		
iOS	4	6	157	2
		7	75	.
		8	523	7
		9	268	1
		10	1356	10
	5	1	3746	28
		2	737	5
		3	858	2
		4	597	10
		5	368	3
		6	184	1
		7	61	.
		8	279	2
		9	126	1
		10	1085	7
	6	2	4882	31
		5	543	2
		7	164	.
		8	333	3
		9	182	1
		10	1674	9
	7	1	2562	38
		3	576	9
		4	1249	17
		6	589	13
		10	2153	29
	8	1	1965	16
		2	455	2
		3	225	.
		4	565	11
		5	258	2
		6	70	.
		8	219	3
		9	86	2

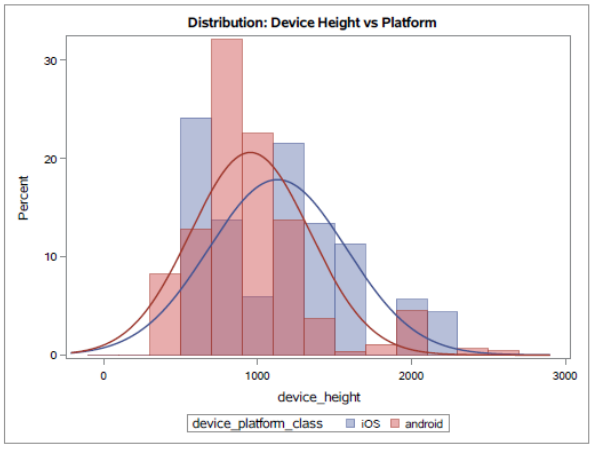
Table: Platform, Make & OS Version vs Install

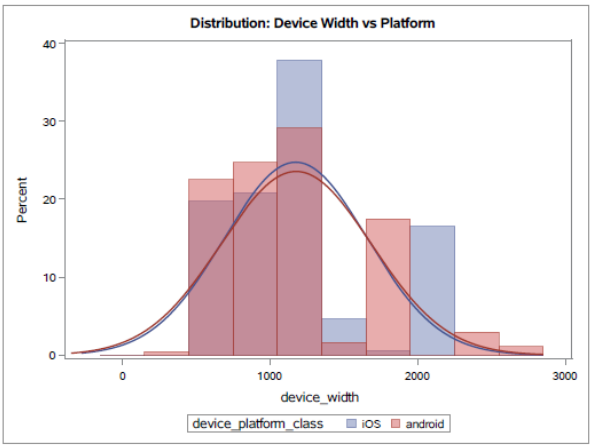
			install	
			0	1
			N	N
device_platform_class	device_make_class	device_os_class		
iOS	8	10	396	12
	9	1	922	13
		2	1287	15
		3	93	1
		4	210	2
		5	229	3
		6	49	1
		7	77	1
		8	147	2
		9	60	.
		10	629	6
	10	1	544	3
		2	1211	21
		3	54	.
		4	172	1
		5	162	5
		6	35	.
		7	3006	18
		8	77	3
		9	77	1
		10	769	1

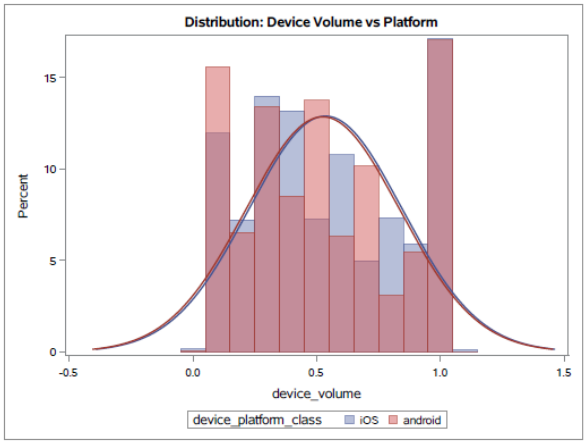
This might force us to model for iOS and Android devices separately. Or develop one model that includes both the data but restrict predictions for android devices with class 10 manufacturer and class 10 publisher app. But one needs to look for other device characteristics to make a decision on both the approaches. The selection is explained in detail in the next section.

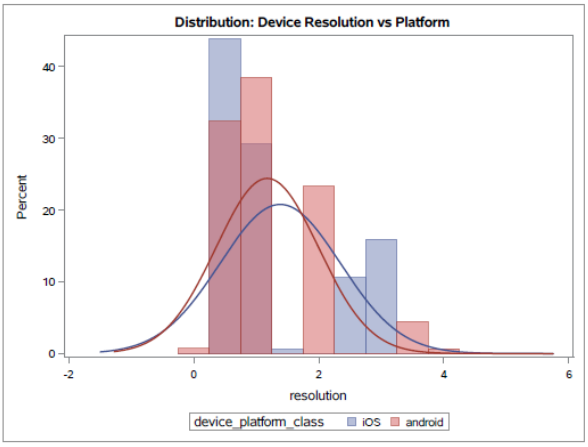
Continuous Variables:

The below plots show distribution of device height, width, volume and resolution for Android and iOS devices. Coinciding with the general intuition, the distribution is remarkably similar, or in other words, they are not significantly different. Hence, the second approach of developing a common model for both the platforms can be chosen and then restricting the prediction for android users with class 10 manufacturer and class 10 publisher app.









Correlation between Device Height and Width with Resolution:

Another interesting observation is that the device height and width are highly correlated with resolution with 0.77 and 0.81, respectively. A clear examination of these variables shows that device height and width are related to resolution with the following formula:

Resolution = (Height * Width) /1,000,000

Pearson Correlation Coefficients, N = 121339 Prob > |r| under H0: Rho=0				
	device_height	device_width	device_volume	resolution
device_height	1.00000	0.26162 <.0001	-0.01301 <.0001	0.76585 <.0001
device_width	0.26162 <.0001	1.00000	0.00199 0.4883	0.81404 <.0001
device_volume	-0.01301 <.0001	0.00199 0.4883	1.00000	-0.01071 0.0002
resolution	0.76585 <.0001	0.81404 <.0001	-0.01071 0.0002	1.00000

By including all three variables, the models run into multi-collinearity. By including resolution, the variance inflation factor is 437.7 and the largest condition index is 48.42. Therefore, resolution is dropped in all the models. Device height and width is used instead which provides better variation compared to resolution.

Model Selection Process

The data is converted from long format to wide format with dummy variable encoding for categorical variables. And for this section (Model Selection Process), the data with dummy variables (original dataset in wide format) is split in 70-30 to create training and test data samples.

Training sample: 84,938 records Test sample: 36,401 records

Linear Probability Models

Model 1: Linear Probability Model with all Variables

All the 33 variables including 9 dummy variables each for manufacturer, OS and publisher app are included in the model. By not including device resolution, the problem of multi-collinearity is resolved. Variation inflation factor for all the variables is less than 10 and the largest condition index is 3.42. However, residuals are not normal and homoscedastic, and one can observe two peaks for each value of install. The linear probability model cannot be taken literally.

The area under curve of ROC for the test sample is 0.6080.

Model 2: Stepwise Linear Probability Model with SBC Selection and best Validation model

Bayesian information criterion penalizes additional variables more than the Akaike information criterion. Hence SBC is chosen as a selection criterion with a more conservative stepwise method. However, the idea of the project is to develop the model with best out-of-sample prediction performance. Therefore, it is imperative to choose the best model which leads to lowest validation average squared error (10% of train sample). This resulted in choosing only three variables, namely – Publisher ID Class 3, Device Make ID Class 7 and Device Height.

The area under curve of ROC for the test sample is 0.5867. The test data performance of this model is comparatively worse than the previous model.

Model 3: Lasso Linear Probability Model with best Validation model

As seen in Model-2, the Bayesian information criterion had heavily penalized the selection of variables to compensate for best significant in-sample fit. This resulted in poor test sample performance. Lasso Regression is another technique which imposes penalty on size of coefficients while estimating model. This resulted in lower prediction errors in out-of-sample data. The chosen best model has the lowest average squared error on validation sample (10 % of train sample).

Even after implementing strict measures on the size of coefficients, this model has included all the variables similar to Model-1. It is a good starting point in variable selection.

Logistic Regression Models

Model 1: Logistic Regression with all Variables

Logistic Regression models have an S-shaped probability function which is reasonable in predicting binary response variable. In logistic regression, logarithm of odds of an event is linear in model parameters. Hence, the predicted probabilities are easily interpreted.

After running the logistic regression, the area under curve of ROC for the test sample is 0.6083. There is a very slight improvement in test performance compared to the linear probability models. But this is not significant.

Model 2: Stepwise Logistic Regression with SBC Selection and best Validation model

Similar to the stepwise linear probability model, the stepwise logistic regression with Bayesian information criterion has penalized heavily in the variable selection. The model selection is done using PROC HPLOGISTIC because the primitive PROC LOGISTIC does not provide such features. Again here, the variables selected are Publisher ID Class 3, Device Make ID Class 7 and Device Height. All the variables are positive and highly significant. This reiterates the fact that advertisements on class 3 publisher app and class 7 iOS manufacturer with greater device height lead to more app installs.

The area under curve of ROC for the test sample is 0.5869 which is a slight improvement compared to its counterpart linear probability model. However, this improvement is not significant at all.

Final Models

For the final models, all the variables are chosen as it results in best test sample performance. However, as discussed earlier in the EDA, the above models can be biased because of rare outcomes. To solve this, the dataset is oversampled to randomly select subset of observations, with higher probability of choosing observations with event (install = 1). For the final models, the oversampled data contains approximately 90% of event (install = 0) and 10% of event (install = 1).

Frequency Counts of Over-Sampled Dataset

The FREQ Procedure

install	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	8620	89.53	8620	89.53
1	1008	10.47	9628	100.00

Running logistic models on the oversampled dataset might lead to less biased maximum likelihood estimates. But the intercept can still be biased since the base probability of event (install = 1) has increased. There are two ways in which this issue can addressed. First is the weight-adjusted model and second is the offset-adjusted model. Both are discussed in detail with the model results in the following sub-sections.

And for the following sub-sections and next section, the oversampled data with dummy variables (oversampled dataset in wide format) is split in 70-30 to create training and test data samples.

Training sample: 6,740 records Test sample: 2,888 records

To compare all the final models, both linear and logistic regression models with all predictors are run on the oversampled dataset.

Final Model 1: Linear Probability Model

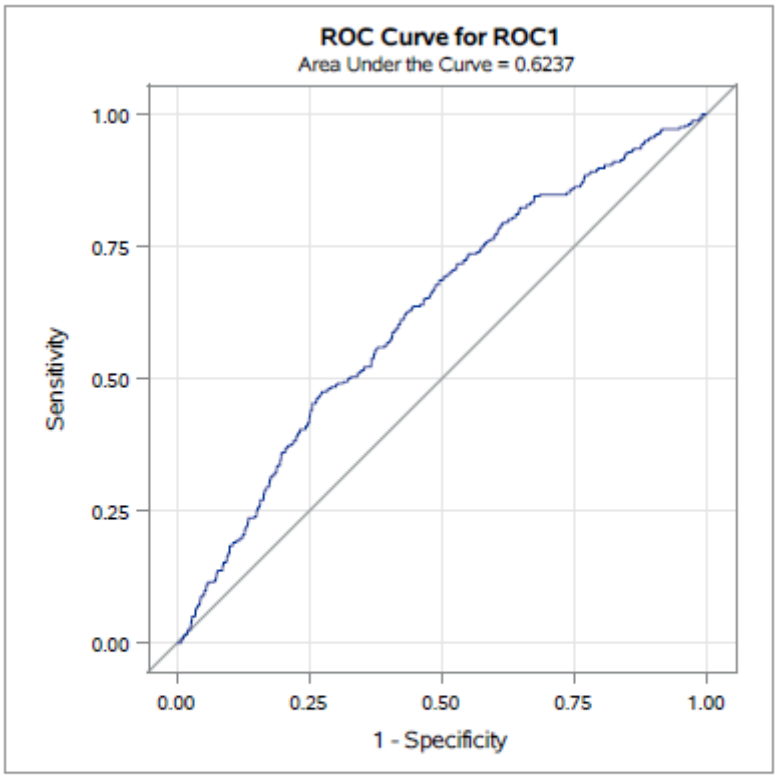
Following are the model estimates by running a linear probability model:

Parameter Estimates						
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	B	0.01165	0.03460	0.34	0.7364
pubid_1	Publisher Id Class 1	B	0.18518	0.10650	1.74	0.0821
pubid_2	Publisher Id Class 2	B	0.02938	0.01480	1.98	0.0472
pubid_3	Publisher Id Class 3	B	0.08119	0.01452	5.59	<.0001
pubid_4	Publisher Id Class 4	B	-0.00337	0.01533	-0.22	0.8263
pubid_5	Publisher Id Class 5	B	0.00971	0.01684	0.58	0.5643
pubid_6	Publisher Id Class 6	B	-0.03589	0.01990	-1.80	0.0713
pubid_7	Publisher Id Class 7	B	-0.04873	0.01924	-2.53	0.0114
pubid_8	Publisher Id Class 8	B	-0.04110	0.02040	-2.02	0.0439
pubid_9	Publisher Id Class 9	B	-0.03015	0.02106	-1.43	0.1523
pubid_10	Publisher Id Class 10	0	0	.	.	.
os_1	Device OS Class 1	B	0.02749	0.01321	2.08	0.0375
os_2	Device OS Class 2	B	0.02024	0.01597	1.27	0.2050
os_3	Device OS Class 3	B	0.00917	0.01722	0.53	0.5945
os_4	Device OS Class 4	B	0.03539	0.01681	2.11	0.0353
os_5	Device OS Class 5	B	0.03932	0.02113	1.86	0.0628
os_6	Device OS Class 6	B	0.04178	0.02351	1.78	0.0756
os_7	Device OS Class 7	B	-0.01089	0.03202	-0.34	0.7339
os_8	Device OS Class 8	B	0.02717	0.02399	1.13	0.2574
os_9	Device OS Class 9	B	0.00831	0.03210	0.26	0.7956
os_10	Device OS Class 10	0	0	.	.	.
plat_ios	Device Platform Class iOS	B	0.02567	0.03763	0.68	0.4952
plat_android	Device Platform Class Android	0	0	.	.	.
make_1	Device Make Class 1	B	-0.04793	0.02372	-2.02	0.0434
make_2	Device Make Class 2	B	-0.06171	0.02393	-2.58	0.0099
make_3	Device Make Class 3	B	-0.03110	0.02556	-1.22	0.2238
make_4	Device Make Class 4	B	-0.07187	0.02795	-2.57	0.0102
make_5	Device Make Class 5	B	-0.03714	0.02643	-1.41	0.1600
make_6	Device Make Class 6	B	-0.06565	0.02625	-2.50	0.0124

make_7	Device Make Class 7	B	0.01147	0.02673	0.43	0.6680
make_8	Device Make Class 8	B	-0.03408	0.03193	-1.07	0.2859
make_9	Device Make Class 9	B	-0.05939	0.03152	-1.88	0.0596
make_10	Device Make Class 10	0	0	.	.	.
wifi		1	0.02585	0.00830	3.11	0.0019
device_height		1	0.00004475	0.00001169	3.83	0.0001
device_width		1	0.00000660	0.00001069	0.62	0.5369
device_volume		1	0.01326	0.01198	1.11	0.2682

The BLUE estimates are biased as explained before. The area under the curve of ROC for the test sample is 0.6237 with 95% confidence interval of 0.5920 – 0.6555.

ROC Model: ROC1



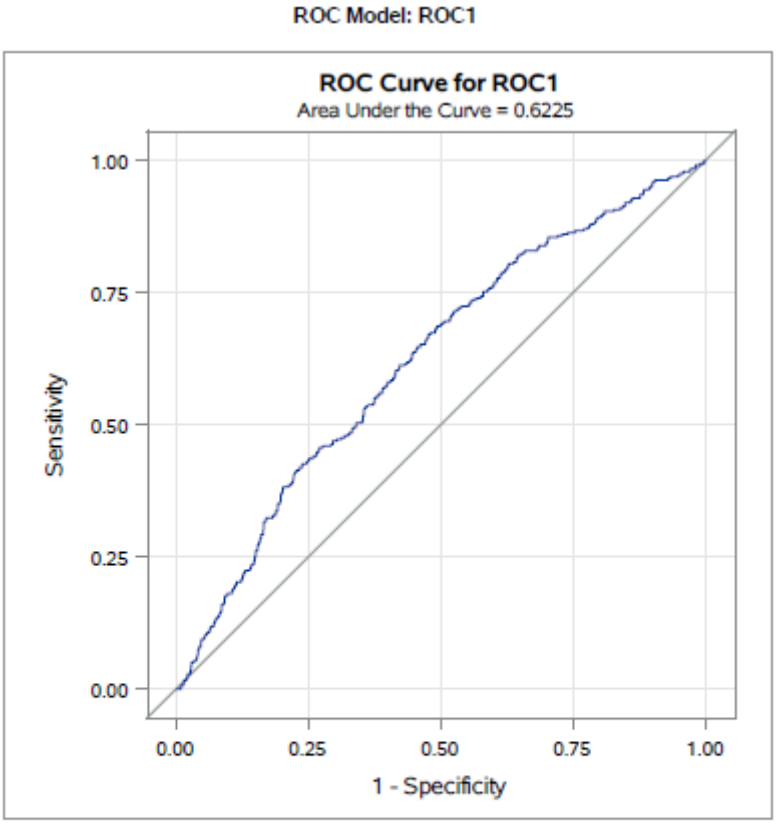
ROC Association Statistics							
	Mann-Whitney						
ROC Model	Area	Standard Error	95% Wald Confidence Limits		Somers' D	Gamma	Tau-a
ROC1	0.6237	0.0162	0.5920	0.6555	0.2474	0.2475	0.0490

Final Model 2: Logistic Regression (Unadjusted Model)

Following are the model estimates by running a simple logistic regression without adjusting the intercept:

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-3.1648	0.3982	63.1808	<.0001
pubid_1	1	1.4978	0.8286	3.2673	0.0707
pubid_2	1	0.3169	0.1609	3.8799	0.0489
pubid_3	1	0.7082	0.1373	26.6111	<.0001
pubid_4	1	-0.0609	0.1859	0.1071	0.7435
pubid_5	1	0.1014	0.1936	0.2741	0.6006
pubid_6	1	-0.6508	0.3073	4.4859	0.0342
pubid_7	1	-0.6250	0.2556	5.9800	0.0145
pubid_8	1	-0.5817	0.2828	4.2314	0.0397
pubid_9	1	-0.3402	0.2522	1.8190	0.1774
pubid_10	0	0	.	.	.
os_1	1	0.3264	0.1577	4.2827	0.0385
os_2	1	0.2675	0.1930	1.9205	0.1658
os_3	1	0.0975	0.2106	0.2143	0.6434
os_4	1	0.3855	0.1862	4.2884	0.0384
os_5	1	0.4612	0.2388	3.7309	0.0534
os_6	1	0.4442	0.2491	3.1796	0.0746
os_7	1	-0.0172	0.3667	0.0022	0.9626
os_8	1	0.3377	0.2784	1.4712	0.2252
os_9	1	0.0942	0.3984	0.0560	0.8130
os_10	0	0	.	.	.
plat_ios	1	0.1771	0.4206	0.1772	0.6738
plat_android	0	0	.	.	.
make_1	1	-0.4304	0.2376	3.2808	0.0701
make_2	1	-0.6023	0.2432	6.1351	0.0133
make_3	1	-0.2499	0.2635	0.8996	0.3429
make_4	1	-0.6577	0.2743	5.7487	0.0165
make_5	1	-0.3202	0.2783	1.3237	0.2499
make_6	1	-0.7311	0.2971	6.0556	0.0139
make_7	1	0.1221	0.2601	0.2203	0.6388
make_8	1	-0.3297	0.3020	1.1914	0.2751
make_9	1	-0.5429	0.3103	3.0608	0.0802
make_10	0	0	.	.	.
wifi	1	0.3123	0.0976	10.2314	0.0014
device_height	1	0.000454	0.000122	13.9267	0.0002
device_width	1	0.000055	0.000115	0.2317	0.6303
device_volume	1	0.1554	0.1329	1.3677	0.2422

The predictor MLE estimates are less biased, but the intercept is still biased. The area under the curve of ROC for the test sample is 0.6225 with 95% confidence interval of 0.5905 – 0.6545. The area under the curve is lower than the linear probability model, but it is not significantly different.



ROC Association Statistics							
	Mann-Whitney						
ROC Model	Area	Standard Error	95% Wald Confidence Limits		Somers' D	Gamma	Tau-a
ROC1	0.6225	0.0163	0.5905	0.6545	0.2450	0.2451	0.0486

Final Model 3: Logistic Regression (Weight-Adjusted Model)

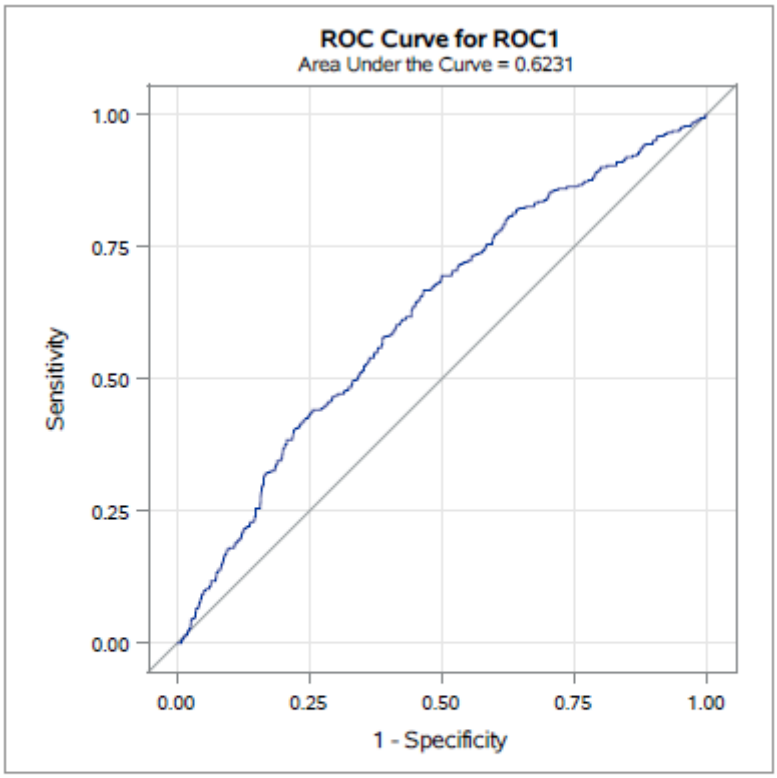
To adjust by weighting, a variable is added to the original dataset that takes the value p1/r1 in event observations, and the value (1-p1)/(1-r1) in nonevent observations, where p1 is the probability of an event in the population and r1 is the proportion of events in your data set.

Following are the model estimates by running a logistic regression by weight adjustment:

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-5.7617	1.3434	18.3943	<.0001
pubid_1	1	1.4990	2.5709	0.3400	0.5599
pubid_2	1	0.3030	0.5393	0.3157	0.5742
pubid_3	1	0.6978	0.4477	2.4287	0.1191

pubid_4	1	-0.0619	0.6313	0.0096	0.9219
pubid_5	1	0.0929	0.6533	0.0202	0.8870
pubid_6	1	-0.6621	1.0625	0.3884	0.5331
pubid_7	1	-0.6400	0.8746	0.5355	0.4643
pubid_8	1	-0.5848	0.9689	0.3643	0.5461
pubid_9	1	-0.3353	0.8544	0.1540	0.6947
pubid_10	0	0	.	.	.
os_1	1	0.3305	0.5330	0.3844	0.5352
os_2	1	0.2668	0.6510	0.1680	0.6819
os_3	1	0.1012	0.7142	0.0201	0.8873
os_4	1	0.3795	0.6225	0.3716	0.5421
os_5	1	0.4567	0.8021	0.3241	0.5691
os_6	1	0.4546	0.8283	0.3012	0.5832
os_7	1	-0.0147	1.2281	0.0001	0.9904
os_8	1	0.3599	0.9387	0.1470	0.7014
os_9	1	0.1054	1.3543	0.0061	0.9380
os_10	0	0	.	.	.
plat_ios	1	0.1688	1.4097	0.0143	0.9047
plat_android	0	0	.	.	.
make_1	1	-0.4136	0.7750	0.2847	0.5936
make_2	1	-0.5885	0.7953	0.5476	0.4593
make_3	1	-0.2486	0.8702	0.0816	0.7751
make_4	1	-0.6248	0.8829	0.5008	0.4792
make_5	1	-0.3159	0.9229	0.1172	0.7321
make_6	1	-0.7118	1.0011	0.5055	0.4771
make_7	1	0.1498	0.8432	0.0315	0.8590
make_8	1	-0.2888	0.9681	0.0890	0.7655
make_9	1	-0.4886	1.0062	0.2358	0.6272
make_10	0	0	.	.	.
wifi	1	0.3170	0.3296	0.9250	0.3362
device_height	1	0.000436	0.000402	1.1783	0.2777
device_width	1	0.000036	0.000388	0.0086	0.9259
device_volume	1	0.1425	0.4435	0.1032	0.7480

The coefficients of the varaibles have not changed but the intercept has reduced (-5.7617) which is necessary to offset the increase in base probability of event (install = 1). The area under the curve of ROC for the test sample is 0.6231 with 95% confidence interval of 0.5911 – 0.6550. However, there is no significant improvement in the test sample performance.



ROC Association Statistics							
	Mann-Whitney						
ROC Model	Area	Standard Error	95% Wald Confidence Limits		Somers' D	Gamma	Tau-a
ROC1	0.6231	0.0163	0.5911	0.6550	0.2461	0.2462	0.0488

Final Model 4: Logistic Regression (Offset-Adjusted Model)

To adjust by using an offset, a variable is added to the original dataset that is defined as log[(r1*(1-p1)) / ((1-r1)*p1)], where log represents the natural logarithm.

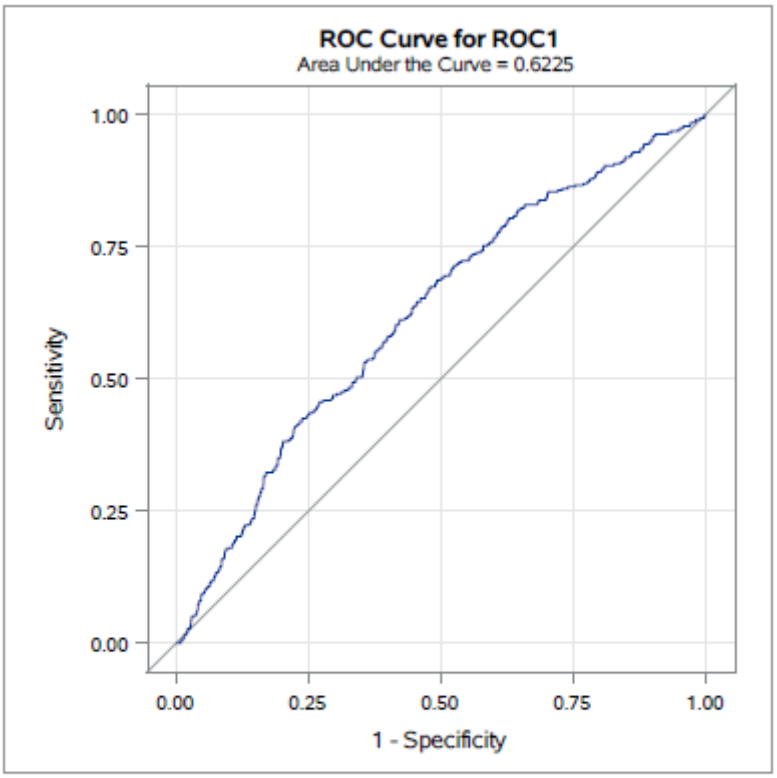
Following are the model estimates by running a logistic regression by weight adjustment:

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-5.8009	0.3981	212.3010	<.0001
pubid_1	1	1.4971	0.8288	3.2634	0.0708
pubid_2	1	0.3168	0.1609	3.8791	0.0489
pubid_3	1	0.7081	0.1373	26.6036	<.0001
pubid_4	1	-0.0609	0.1859	0.1072	0.7433
pubid_5	1	0.1013	0.1936	0.2739	0.6007

pubid_6	1	-0.6514	0.3074	4.4925	0.0340
pubid_7	1	-0.6250	0.2556	5.9803	0.0145
pubid_8	1	-0.5818	0.2828	4.2315	0.0397
pubid_9	1	-0.3401	0.2522	1.8190	0.1774
pubid_10	0	0	.	.	.
os_1	1	0.3264	0.1577	4.2827	0.0385
os_2	1	0.2675	0.1930	1.9206	0.1658
os_3	1	0.0975	0.2106	0.2143	0.6434
os_4	1	0.3855	0.1862	4.2883	0.0384
os_5	1	0.4613	0.2388	3.7310	0.0534
os_6	1	0.4442	0.2491	3.1794	0.0746
os_7	1	-0.0171	0.3667	0.0022	0.9627
os_8	1	0.3377	0.2784	1.4712	0.2252
os_9	1	0.0942	0.3984	0.0560	0.8130
os_10	0	0	.	.	.
plat_ios	1	0.1769	0.4206	0.1769	0.6741
plat_android	0	0	.	.	.
make_1	1	-0.4303	0.2376	3.2807	0.0701
make_2	1	-0.6023	0.2432	6.1348	0.0133
make_3	1	-0.2499	0.2635	0.8994	0.3429
make_4	1	-0.6579	0.2743	5.7514	0.0165
make_5	1	-0.3202	0.2783	1.3235	0.2500
make_6	1	-0.7311	0.2971	6.0561	0.0139
make_7	1	0.1221	0.2601	0.2202	0.6389
make_8	1	-0.3298	0.3020	1.1926	0.2748
make_9	1	-0.5430	0.3103	3.0626	0.0801
make_10	0	0	.	.	.
wifi	1	0.3123	0.0976	10.2306	0.0014
device_height	1	0.000454	0.000122	13.9291	0.0002
device_width	1	0.000056	0.000115	0.2330	0.6293
device_volume	1	0.1554	0.1329	1.3675	0.2422
off	0	1.0000	0	.	.

Similar to the weight-adjusted logistic regression model, the coefficients of the predictors in this offset-adjusted model have not changed but the intercept has reduced (-5.8009). The area under the curve of ROC for the test sample is 0.6225 with 95% confidence interval of 0.5905 – 0.6545. Even this model is not significantly different to all other models in terms of test sample performance.

ROC Model: ROC1



ROC Association Statistics							
	Mann-Whitney						
ROC Model	Area	Standard Error	95% Wald Confidence Limits		Somers' D	Gamma	Tau-a
ROC1	0.6225	0.0163	0.5905	0.6545	0.2450	0.2451	0.0486

Verdict:

Logistic regression modelling for rare outcomes has not particularly improved the performance on test samples compared to the linear probability model (biased estimates). But the intercept adjustment led to unbiased MLE estimates. Both weight-adjusted and offset-adjusted models perform similarly and can be used to calculate the total costs.

Total Costs of Final Models

The advertising platform would like to determine whether to show the ad depending on the publisher and consumer characteristics. In particular, the advertising platform needs to come up with a threshold such that if the probability of installing the ad is above that threshold, the ad is shown to the consumer. This is necessary to minimize the estimated advertisement costs. According to the advertiser, showing an ad to a consumer who would not install the app results in some inconvenience cost to the consumer which in turn leads to less participation and causes a loss of 1 cent (False Positives) to the platform. On the other hand, not showing an ad to a consumer who would have installed the app results in a missed opportunity cost of 100 cents (False Negatives) to the platform.

By plugging the ROC data of the final models, the minimum estimated costs of each model can be derived.

Final Model 1: Linear Probability Model

falpos_001	2565	falpos_020	2520	falpos_040	2450
falneg_001	0	falneg_020	4	falneg_040	8
cost_001	2565	cost_020	2920	cost_040	3250
falpos_005	2560	falpos_025	2507	falpos_045	2399
falneg_005	0	falneg_025	4	falneg_045	9
cost_005	2560	cost_025	2907	cost_045	3299
falpos_010	2550	falpos_030	2490	falpos_050	2335
falneg_010	0	falneg_030	6	falneg_050	12
cost_010	2550	cost_030	3090	cost_050	3535
falpos_015	2542	falpos_035	2472		
falneg_015	2	falneg_035	7		
cost_015	2742	cost_035	3172		

The least estimated cost is $25.61 which occurs at probability threshold = 0.010 on a test sample of 2,888. False Positives are 2,550 and False Negatives are 0.

Final Model 2: Logistic Regression (Unadjusted Model)

Obs	_PROB_	_POS_	_NEG_	_FALPOS_	_FALNEG_	_SENSIT_	_1MSPEC_	total_cost
1	0.028586	322	5	2561	0	1	0.99805	2561

The least estimated cost is $25.61 which occurs at probability threshold = 0.028586 on a test sample of 2,888.

Final Model 3: Logistic Regression (Weight-Adjusted Model)

Obs	_PROB_	_POS_	_NEG_	_FALPOS_	_FALNEG_	_SENSIT_	_1MSPEC_	total_cost
1	.002102956	322	5	2561	0	1	0.99805	2561

The least estimated cost is $25.61 which occurs at probability threshold = 0.02102956 on a test sample of 2,888.

Final Model 4: Logistic Regression (Offset-Adjusted Model)

Obs	_PROB_	_POS_	_NEG_	_FALPOS_	_FALNEG_	_SENSIT_	_1MSPEC_	total_cost
1	0.028568	322	5	2561	0	1	0.99805	2561

The least estimated cost is $25.61 which occurs at probability threshold = 0.028568 on a test sample of 2,888.

Verdict:

There are two insights from the final models and the total costs.

Firstly, the linear probability model has slightly higher AUC and lower total cost compared to both the adjusted models, but this AUC is not significantly different. The linear probability model is biased, and the direct interpretation of the probabilities do not make much sense. This reiterates in favor of adjusted logistic regression models.

Secondly, from the total costs, one can observe that the models predict very few True Negatives and therefore the specificity rate is very low. This is due to the added weightage given to correctly predicting the prospective consumers.

Conclusion

The app install dataset has two issues to deal with – namely model selection and rare outcomes. In terms of model selection, PROC GLMSELECT and PROC HPLOGISTIC selected the entire set of variables based on validation ASE. Class 3 publication app and class 7 manufacturer with greater display size are favorable predictors for app installations. On the other hand, rare outcomes are quite common in real life. Examples include predicting fraud transactions, epidemiologic studies of rare diseases, e-commerce click rates. Running a simple linear regression is not the best way to model binary variables as they are not directly interpretable. And running a simple logistic regression would lead to biased MLE estimates. Therefore, oversampling the lesser class provides a much-needed solution. By using weight-adjusted or offset-adjusted intercepts, the estimates become unbiased. In the case of app install, all the models fare similarly in classifying and minimizing the total costs. Therefore, the adjusted models are preferably used for prediction. For the app developer, showcasing the advertisement for all consumers would be a better approach because of the added weightage given to predicting prospective consumers. This shows that the selected model has exceptionally low specificity. To develop a better classifier, it is important to collect new independent variables or collect more data.

References

- https://documentation.sas.com/?docsetId=stathpug&docsetTarget=stathpug_hplogistic_overview01.htm&docsetVersion=15.1&locale=en
- https://support.sas.com/kb/22/601.html
- https://www.eio.upc.edu/ca/seminari/docs/georg-heinze-logistics-regression-with-rare-events-problems-and-solutions.pdf