Analysis of Mobile Advertising with SAS

By

Sinduja Senthil Kumar

Dataset: -

The dataset contains details about various publisher and consumer characteristics based on which an app was installed. Now the app developer would like to choose the optimal payment based on the number of installs. Hence, our goal is to estimate the probability of installing the ad.

Reading the dataset:

The dataset "DATA" is read using the data step and named as "Advertise". There were 121339 observations read from the data set PROJ.DATA with 10 variables. The variable "install" indicates whether the app was installed or not. Thus we understand that "install" has only two outcomes and thus it is a **Binary Classification Model**

TABLE ADVERTISE:

	install	device_volume	wifi	resolution	device_height	device_width	publisher_id_class	device_os_class	device_make_class	device_platform_class
1	0	0.870000005	1	0.727039993	640	1136	3	1	1	iOS
2	0	0.860000014	1	1.000499964	750	1334	10	4	2	iOS
3	0	0.560000002	1	0.727039993	1136	640	10	1	5	iOS
4	0	1	1	0.727039993	640	1136	10	4	5	iOS
5	0	0.119999997	1	0.727039993	640	1136	6	3	1	iOS
6	0	1	1	0.727039993	640	1136	10	1	1	iOS
7	0	0.310000002	0	0.727039993	1136	640	9	10	3	iOS
8	0	0.469999999	0	2.251125097	2001	1125	10	1	2	iOS
9	0	0.119999997	0	0.727039993	1136	640	8	5	2	iOS
10	0	0.059999999	1	0.727039993	1136	640	3	1	3	iOS
11	0	0.189999998	1	0.727039993	1136	640	3	4	1	iOS
12	0	0.810000002	0	0.727039993	640	1136	6	1	1	iOS
13	0	0.439999998	1	0.727039993	1136	640	10	1	3	iOS
14	0	0.310000002	1	0.727039993	1136	640	7	1	2	iOS
15	0	0.119999997	1	0.727039993	640	1136	6	5	1	iOS
16	0	0.460000008	0	0.727039993	640	1136	3	5	2	iOS

Splitting of dataset:

When building prediction models, it is essential to build the model based on the training dataset and then make predictions based on the test data. This is done in order to avoid overfitting of the model. The dataset "advertise" into training and test with 70% of the observations in training dataset and the remaining 30% observations in the test dataset. This is done using the **PROC surveyselect** command. The output table is as below:

TABLE ADVERTISE WITH INDICATORS FOR TEST AND TRAIN DATA:

	Selection Indicator	install	device_volume	wifi	resolution	device_height	device_width	publisher_id_class	device_os_class	device_make_class	devi
1	0	0	0.870000005	1	0.727039993	640	1136	3	1	1	iOS
2	1	0	0.860000014	1	1.000499964	750	1334	10	4	2	iOS
3	1	0	0.560000002	1	0.727039993	1136	640	10	1	5	iOS
4	0	0	1	1	0.727039993	640	1136	10	4	5	iOS
5	0	0	0.119999997	1	0.727039993	640	1136	6	3	1	iOS
6	0	0	1	1	0.727039993	640	1136	10	1	1	iOS
7	1	0	0.310000002	0	0.727039993	1136	640	9	10	3	iOS
8	1	0	0.469999999	0	2.251125097	2001	1125	10	1	2	iOS
9	1	0	0.119999997	0	0.727039993	1136	640	8	5	2	iOS
10	0	0	0.059999999	1	0.727039993	1136	640	3	1	3	iOS
11	1	0	0.189999998	1	0.727039993	1136	640	3	4	1	iOS
12	1	0	0.810000002	0	0.727039993	640	1136	6	1	1	iOS
13	1	0	0.439999998	1	0.727039993	1136	640	10	1	3	iOS
14	1	0	0.310000002	1	0.727039993	1136	640	7	1	2	iOS
15	0	0	0.119999997	1	0.727039993	640	1136	6	5	1	iOS
16	0	0	0.460000008	0	0.727039993	640	1136	3	5	2	iOS
17	1	0	0.430000007	0	0.727039993	640	1136	10	2	2	iOS
18	1	0	0.439999998	1	2.742336035	1242	2208	5	3	1	iOS
19	1	0	1	1	2.742336035	1242	2208	10	1	1	iOS
20	1	0	0.310000002	0	0.727039993	1136	640	10	1	1	iOS
21	0	0	0.870000005	1	3.145728111	2048	1536	10	4	4	iOS

The observations with selection indicator “1” belong to training dataset and “0” belongs to test dataset. Also, individual datasets for training and test were created and named as “ad_training” and “ad_test”.

Part I.

Linear probability model:

Based on the dataset we understand that it is a classification problem with the dependent variable just having values 0 and 1. However, we are more interested in predicting the probabilities of install being 0 or 1, rather than just making class predictions. Since Linear probability models can make probability predictions we begin with this approach.

Initial Model - Linear probability model:

In the initial model all predictors from the dataset are added. Linear Probability model can be run using PROC REG step. However, there cannot be any categorical indicators in the dataset. From the table “advertise” we see that there is one categorical predictor present in the dataset which is “device_platform_class”. Hence by using glm_mod we generate indicator variables for this categorical variable.

<u>TABLE ADVERTISE WITH INDICATORS FOR CATEGORICAL VARIABLE “device_platform_class”:</u>

	install	Selection Indicator	device_volume	wifi	resolution	device_height	device_width	publisher_id_class	device_os_class	device_make_class	device_platform_clas android	device_platform_clas iOS
1	0	0	0.870000005	1	0.727039993	640	1136	3	1	1	0	1
2	0	1	0.860000014	1	1.000499964	750	1334	10	4	2	0	1
3	0	1	0.560000002	1	0.727039993	1136	640	10	1	5	0	1
4	0	0	1	1	0.727039993	640	1136	10	4	5	0	1
5	0	0	0.119999997	1	0.727039993	640	1136	6	3	1	0	1
6	0	0	1	1	0.727039993	640	1136	10	1	1	0	1
7	0	1	0.310000002	0	0.727039993	1136	640	9	10	3	0	1
8	0	1	0.469999999	0	2.251125097	2001	1125	10	1	2	0	1
9	0	1	0.119999997	0	0.727039993	1136	640	8	5	2	0	1
10	0	0	0.059999999	1	0.727039993	1136	640	3	1	3	0	1
11	0	1	0.189999998	1	0.727039993	1136	640	3	4	1	0	1
12	0	1	0.810000002	0	0.727039993	640	1136	6	1	1	0	1
13	0	1	0.439999998	1	0.727039993	1136	640	10	1	3	0	1
14	0	1	0.310000002	1	0.727039993	1136	640	7	1	2	0	1
15	0	0	0.119999997	1	0.727039993	640	1136	6	5	1	0	1
16	0	0	0.460000008	0	0.727039993	640	1136	3	5	2	0	1
17	0	1	0.430000007	0	0.727039993	640	1136	10	2	2	0	1
18	0	1	0.439999998	1	2.742336035	1242	2208	5	3	1	0	1
19	0	1	1	1	2.742336035	1242	2208	10	1	1	0	1
20	0	1	0.310000002	0	0.727039993	1136	640	10	1	1	0	1
21	0	0	0.870000005	1	3.145728111	2048	1536	10	4	4	0	1

From the above table we see that the predictor “device_platform_class” is replaced with indicator variables “device_platform_class android” and “device_platform_class iOS”. Now the linear probability model is executed with PROC reg and the results are as below.

From the p values we see that the model is significant. And the p values for the individual predictors show that the predictor device_volume is insignificant and predictor device_platform_class android when compared to predictor device_platform_class iOS is insignificant.

Initial Linear Probability Model - PROC REG results:

The SAS System

The REG Procedure
Model: MODEL1
Dependent Variable: install

Number of Observations Read	84938
Number of Observations Used	84938

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	9	0.53476	0.05942	7.49	<.0001
Error	84928	674.02127	0.00794		
Corrected Total	84937	674.55603			

Root MSE	0.08909	R-Square	0.0008
Dependent Mean	0.00801	Adj R-Sq	0.0007
Coeff Var	1112.76760		

Parameter Estimates						
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	B	-0.01713	0.00676	-2.53	0.0113
Col1	device_volume	1	0.00145	0.00099621	1.46	0.1450
Col2	wifi	1	0.00173	0.00067986	2.55	0.0109
Col3	resolution	1	-0.01281	0.00412	-3.11	0.0019
Col4	device_height	1	0.00001889	0.00000531	3.56	0.0004
Col5	device_width	1	0.00001804	0.00000534	3.38	0.0007
Col6	publisher_id_class	1	-0.00041303	0.00011533	-3.58	0.0003
Col7	device_os_class	1	-0.00015655	0.00010321	-1.52	0.1293
Col8	device_make_class	1	0.00057088	0.00014391	3.97	<.0001
Col9	device_platform_clas android	B	-0.00085320	0.00259	-0.33	0.7420
Col10	device_platform_clas iOS	0	0	.	.	.

Procedure and Measures to choose the Final Linear Probability Model:

It is important to choose the best predictors before we use them in our final model to get the right predictions. Hence, we begin our trials by taking log for the numerical predictors present in our dataset, which were log_device_volume, log_resolution, ldevice_height and log_dev_width.

Trial 1- LOG MODEL:

On running PROC reg one the dataset with new log predictors, the p values of the individual predictors didn't improve from our values obtained from the "INITIAL Model". Hence, we decide to use the predictors without log and stick to our initial model.

Though we have decided to use the non-log initial model, to know which predictors are to be added and removed we use the ITERATIVE APPROACH.

ITERATIVE APPROACHES:

Trial 2- Forward Selection:

Using PROC GLMSELECT we use the forward selection procedure to decide on the predictors that gets added to the model. Initially the model has 0 predictors and based on the selection criteria chosen the predictors get added to the model one by one. This procedure stops executing once the value of the incoming predictor falls below the set criterion value.

For our model, the selection criteria chosen was "Significance Level" and the Significance entry value was set to 0.2. And as a result of the Forward Selection Procedure, 6 out of the 10 predictors were added to the model.

Trial 3 - Backward Selection:

The same procedure as in forward selection method is to be followed. However here the initially model consists of all the predictors and the predictors are dropped one by one based on the set selection criteria value. Here also the selection criteria were chosen to be "Significance Level", however instead of Significance entry value, here Significance exit value is provided which is 0.15. And as a result of the Backward Selection Procedure, 8 out of the 10 predictors were present in the final model.

Trial 4 - Stepwise Selection:

Stepwise Selection procedure is a combination of the Forward and Backward selection. The predictors get added and removed simultaneously based on the set criterion. Significance level was chosen as the selection criterion here and the result of the Stepwise Selection Procedure include 5 out of the 10 predictors.

Trial 5 - Best subsets regression:

Using PROC REG the best subset regression was run to get the 10 best models with right predictors based on various criterion such as Cp, AIC, BIC values. The results from the best subset regression is as below.

BEST 10 MODELS:

Number in Model	C(p)	R-Square	Adjusted R-Square	AIC	BIC	Variables in Model
8	8.1084	0.0008	0.0007	-410777.32	-410775.32	Col1 Col2 Col3 Col4 Col5 Col6 Col7 Col8
7	8.2138	0.0008	0.0007	-410777.22	-410775.22	Col2 Col3 Col4 Col5 Col6 Col7 Col8
6	8.6910	0.0007	0.0007	-410776.74	-410774.74	Col2 Col3 Col4 Col5 Col6 Col8
7	8.6916	0.0008	0.0007	-410776.74	-410774.74	Col1 Col2 Col3 Col4 Col5 Col6 Col8
9	10.0000	0.0008	0.0007	-410775.43	-410773.43	Col1 Col2 Col3 Col4 Col5 Col6 Col7 Col8 Col10
9	10.0000	0.0008	0.0007	-410775.43	-410773.43	Col1 Col2 Col3 Col4 Col5 Col6 Col7 Col8 Col9
8	10.1243	0.0008	0.0007	-410775.31	-410773.31	Col2 Col3 Col4 Col5 Col6 Col7 Col8 Col10
8	10.1243	0.0008	0.0007	-410775.31	-410773.31	Col2 Col3 Col4 Col5 Col6 Col7 Col8 Col9
8	10.3008	0.0008	0.0007	-410775.13	-410773.13	Col1 Col2 Col3 Col4 Col5 Col6 Col8 Col10
8	10.3008	0.0008	0.0007	-410775.13	-410773.13	Col1 Col2 Col3 Col4 Col5 Col6 Col8 Col9

Thus, from the **ITERATIVE APPROACH** and **Best subsets regression** using the criteria - Significance level, BIC and AIC values we see that the best model is the one with 8 predictors that was chosen from the Backward Selection Procedure. And the chosen 8 predictors are WIFI, resolution, device volume, device_height, device_width, publisher_id_class, device_os_class and device_make_class .

However, as it is more appropriate to choose the best model based on the test data instead of the training dataset as that would **avoid the problems of overfitting.** Hence, we further proceed by using the ASE values to compare between the test and train and then decide on the final model that can **generalize** well.

Trial 6 - Backward selection with p-value as criteria (ASE in train vs. test data):

The first step was to create a smaller dataset because using a larger dataset to finalize on these procedures would give more or less the same results. Thus, with a smaller dataset the results from these procedures could be more differentiable. Here from the training sample, 25% of random observations were included for the smaller dataset.

In this trial the backward selection procedure was used with the significance level as the selection criterion and the exit value is set to be 0.15. Also, this model included the interaction terms for all the predictors and the ASE values was compared between the TEST and TRAIN data.

Trial 7 - Backward selection with AIC as criteria (ASE in train vs. test data)

Like the previous trial, this trial is done with backward selection procedure along with interaction terms. However, here the selection criteria is chosen to be AIC instead of the significance level.

Both trial 6 and trial 7 gave similar ASE values for the train and test dataset inspite of using smaller dataset. We see that the ASE for the test data is slightly higher than the training data.

ASE of train and test data from Trial 6 and Trial 7:

Root MSE	0.09070
Dependent Mean	0.00831
R-Square	0.0028
Adj R-Sq	0.0020
AIC	-80448
AICC	-80448
SBC	-101480
ASE (Train)	0.00822
ASE (Test)	0.00893

In trial 6 and 7 we focused on **choosing a metric based on the in-sample dataset and cross checked its performance across the out-sampled data.** In trial 8 and 9 we would focus on using **cross validation** to choose the best model.

Trial 8 - Backward selection with p-value as criteria and validation dataset to choose predictors

The first step here is to set aside a portion of the dataset as validation dataset. From the small sampled dataset 20% of the observations are set aside as validation data. The backward selection procedure is executed with significance level as criterion and all the predictors were chosen based on the validation data. Interaction terms between predictors were also added.

Trial 9 - Backward selection with AIC as criteria and validation dataset to choose predictors

This trial is similar to trial 8, however instead of using the significance level as criterion, AIC values was used. Both trial 8 and trial 9 gave similar ASE values for the train, validate and test dataset. We see that the ASE values for the validation data is higher than the training data here as in these trials' validation dataset was used for selection of predictors and hence the overfit is on the validation data.

After all these trials based on ASE value there are not much difference between these trials and hence we decide to use the 8 predictors from trial 3 in our final model.

ASE of train, validation and test data from Trial 8 and Trial 9:

Root MSE	0.08684
Dependent Mean	0.00762
R-Square	0.0031
Adj R-Sq	0.0023
AIC	-65832
AICC	-65832
SBC	-82657
ASE (Train)	0.00753
ASE (Validate)	0.01099
ASE (Test)	0.00894

Final Model - Linear probability model:

The final model is run using PROC REG with 8 predictors and the results are as below.

Final Linear Probability Model - PROC REG results:

The SAS System

The REG Procedure
Model: Final_Model
Dependent Variable: install

Number of Observations Read	84938
Number of Observations Used	84938

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	8	0.53390	0.06674	8.41	<.0001
Error	84929	674.02213	0.00794		
Corrected Total	84937	674.55603			

Root MSE	0.08909	R-Square	0.0008
Dependent Mean	0.00801	Adj R-Sq	0.0007
Coeff Var	1112.76176		

Parameter Estimates						
Variable	**Label**	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	Intercept	1	-0.01673	0.00665	-2.52	0.0119
Col1	device_volume	1	0.00145	0.00099599	1.45	0.1468
Col2	wifi	1	0.00173	0.00067985	2.54	0.0110
Col3	resolution	1	-0.01260	0.00408	-3.09	0.0020
Col4	device_height	1	0.00001865	0.00000526	3.54	0.0004
Col5	device_width	1	0.00001776	0.00000527	3.37	0.0008
Col6	publisher_id_class	1	-0.00041729	0.00011461	-3.64	0.0003
Col7	device_os_class	1	-0.00016293	0.00010137	-1.61	0.1080
Col8	device_make_class	1	0.00055734	0.00013791	4.04	<.0001

Thus, from the results based on the p values we see that the overall model is significant. Though the linear probability model can give predicted probabilities and hence can be used for binary logit models, there are two major problems with this model.

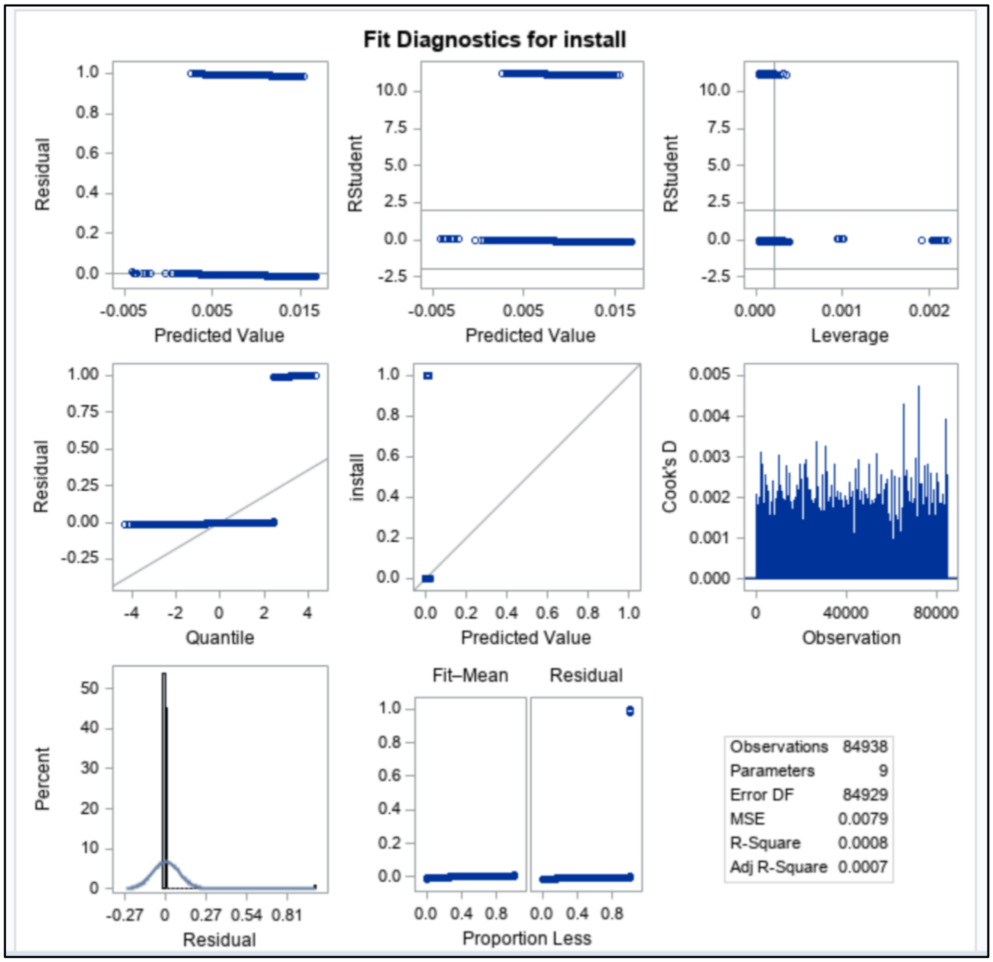
Linear probability model problems:

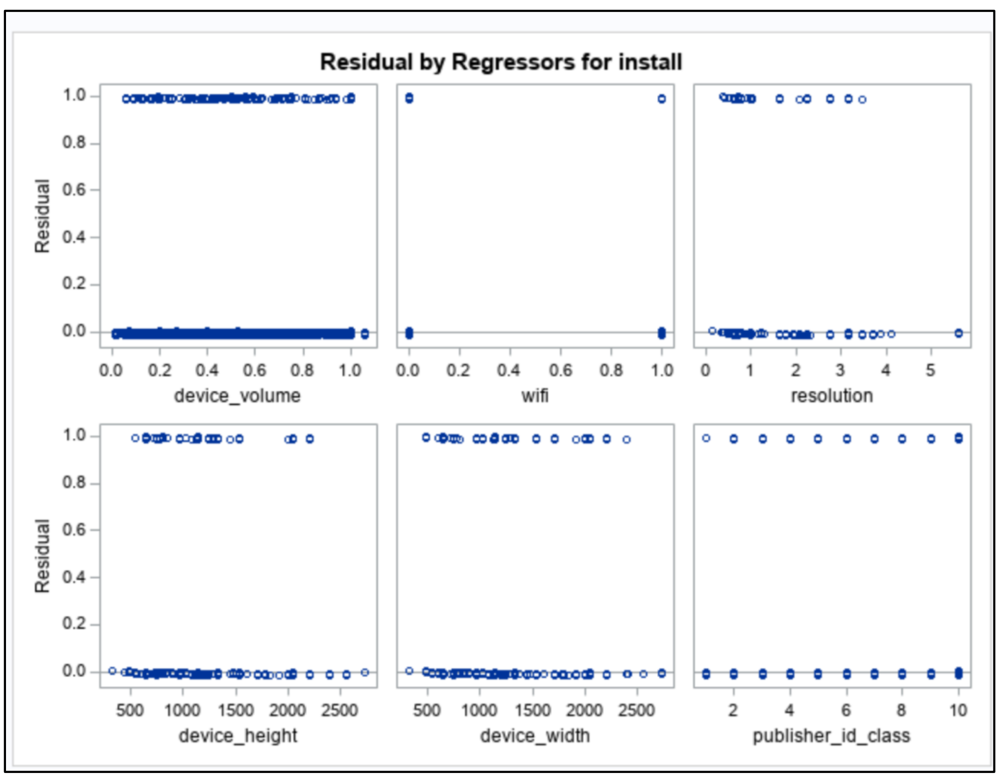
1. The predicted probabilities can have values lesser than 0 and higher than 1
2. Residuals are not normal

In our case we see that we do not have the issue of probabilities lesser than 0 or greater than 1 based on the screenshot below. However, from the **residual plots it is seen that the residuals are not normal**.

Final Linear Probability Model – Probability values within 0 and 1

	install	device_volume	wifi	resolution	device_height	device_width	publisher_id_class	device_os_class	device_make_class	device_platform_clas android	device_platform_clas iOS	Predicted Value of install
1	0	0.860000014	1	1.000499964	750	1334	10	4	2	0	1	0.0076055773
2	0	0.560000002	1	0.727039993	1136	640	10	1	5	0	1	0.0076507119
3	0	0.310000002	0	0.727039993	1136	640	9	10	3	0	1	0.0033962748
4	0	0.469999999	0	2.251125097	2001	1125	10	1	2	0	1	0.0096603143
5	0	0.119999997	0	0.727039993	1136	640	8	5	2	0	1	0.0037962853
6	0	0.189999998	1	0.727039993	1136	640	3	4	1	0	1	0.0073188135
7	0	0.810000002	0	0.727039993	640	1136	6	1	1	0	1	0.0052820199
8	0	0.439999998	1	0.727039993	1136	640	10	1	3	0	1	0.0063625992
9	0	0.310000002	1	0.727039993	1136	640	7	1	2	0	1	0.0068692349
10	0	0.430000007	0	0.727039993	640	1136	10	2	2	0	1	0.0034581133
11	0	0.439999998	1	2.742336035	1242	2208	5	3	1	0	1	0.0114402237
12	0	1	1	2.742336035	1242	2208	10	1	1	0	1	0.0104889736
13	0	0.310000002	0	0.727039993	1136	640	10	1	1	0	1	0.0033306852
14	0	1	1	1.000499964	1334	750	4	6	1	0	1	0.0099469572
15	0	0.310000002	1	3.145728111	1536	2048	6	1	8	0	1	0.01261959
16	0	0.560000002	1	0.727039993	640	1136	2	1	1	0	1	0.00831921


Fit Diagnostics for install
Residual
Predicted Value
RStudent
Predicted Value
RStudent
Leverage
Residual
Quantile
install
Predicted Value
Cook's D
Observation
Percent
Residual
Fit–Mean
Residual
Proportion Less
Observations 84938
Parameters 9
Error DF 84929
MSE 0.0079
R-Square 0.0008
Adj R-Square 0.0007


Residual by Regressors for install
Residual
device_volume
wifi
resolution
Residual
device_height
device_width
publisher_id_class

This violates one of the 4 assumptions of linear regression and since the assumption is violated, the std error estimates will be wrong and hence we cannot decide on the significance of a predictor based on p values. Also, a unit change in X does not have the same impact on probability. Because of these problems we decide to use generate predictions with logit model using PROC LOGISTIC.

Logistic regression model:

Next, we develop the logistic regression model. The initial model is run the same way as the initial linear probability model with all the predictors on the training dataset with indicators.

Initial model - Logistic Regression:

Logistic Regression is run with PROC LOGISTIC with all 10 predictor variables using the training dataset.

Initial Logistic Regression Model - PROC LOGISTIC results:

The SAS System

The LOGISTIC Procedure

Model Information	
Data Set	WORK.AD_TRAINING_WITH_INDICATORS
Response Variable	install
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	84938
Number of Observations Used	84938

Response Profile		
Ordered Value	install	Total Frequency
1	0	84258
2	1	680

Probability modeled is install='1'.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	7922.056	7874.596
SC	7931.405	7968.093
-2 Log L	7920.056	7854.596

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-8.0549	0.8342	93.2452	<.0001
Col1	1	0.1870	0.1252	2.2316	0.1352
Col2	1	0.2423	0.0910	7.0923	0.0077
Col3	1	-1.6480	0.5035	10.7123	0.0011
Col4	1	0.00240	0.000649	13.6580	0.0002
Col5	1	0.00230	0.000654	12.3452	0.0004
Col6	1	-0.0504	0.0143	12.4208	0.0004
Col7	1	-0.0194	0.0133	2.1160	0.1458
Col8	1	0.0682	0.0174	15.4126	<.0001
Col9	1	-0.1397	0.3255	0.1841	0.6679
Col10	0	0	.	.	.

From the p values of chi square test, we see that most of the predictors are significant.

Procedure and Measures to choose the Final Logistic Regression Model:

Following this, we perform the various selection procedures to decide on the predictors to be included in the final model which are summarized in the table as shown below. Here we use PROC LOGISTIC with "selection" option for running the various selection procedures instead of PROC GLM unlike in linear probability model

Logistic Regression Model -Selection procedure trials

Trial	Selection Method	Selection Criteria & Parameters	Number of Predictors	Log-likelihood
1	Stepwise Selection	Significance level: - Entry value: 0.25 Stay value: 0.35	8	7854.786
2	Forward Selection	Significance level: - Entry value: 0.25	8	7854.786
3	Backward Selection	Significance level: - Stay value: 0.35	8	7854.786

From the above results, we observe that the log-likelihood values for the models obtained from all the three selection procedures yield the same value. This value 7854.**786** is higher than the log-likelihood value of 7854.**596** and hence, we are permitted to choose any 1 of the three models. Therefore, as per trial 1, we decide to go with stepwise selection for our final model which has 8 predictors.

Final model - Logistic Regression:

We then run proc logistic on the final model as per our selection procedure and the results are as follows,

Final Logistic Regression Model - PROC LOGISTIC results:

The SAS System

The LOGISTIC Procedure

Model Information	
Data Set	WORK.AD_TRAINING_WITH_INDICATORS
Response Variable	install
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	84938
Number of Observations Used	84938

Response Profile		
Ordered Value	install	Total Frequency
1	0	84258
2	1	680

Probability modeled is install='1'.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	7922.056	7872.786
SC	7931.405	7956.933
-2 Log L	7920.056	7854.786

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-7.9600	0.8020	98.4987	<.0001
Col1	1	0.1859	0.1252	2.2060	0.1375
Col2	1	0.2414	0.0910	7.0441	0.0080
Col3	1	-1.5962	0.4876	10.7181	0.0011
Col4	1	0.00234	0.000632	13.6963	0.0002
Col5	1	0.00223	0.000632	12.4204	0.0004
Col6	1	-0.0511	0.0142	12.9219	0.0003
Col7	1	-0.0207	0.0131	2.5017	0.1137
Col8	1	0.0658	0.0165	15.9057	<.0001

Rare Events:

In the above approach, we do not consider modelling the rare events. This is because the number of rare events (i.e. event=1) is 680 which is reasonably high as per the **Thumb rule** for considering rare events which states that "There should be at least 20 events per independent variable".

Since our model has 10 independent predictors, 10*20 = 200 is the number of observations that the model should have in it's rare category for those events to be considered rare. However, from the results, we observe that there are 680 observations in the rare category of the training dataset (event=1) and hence, the modeling of rare events would generally be not required for this particular model.

Oversampling approach to handle rare events:

- In the first step, we use proc freq to generate a table which displays the count and frequency of both the rare and non-rare events for the full dataset as follows,

Before Oversampling for Rare Events - Count of events '0' and '1'

response counts in full data set

The FREQ Procedure

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

- Next, we create a subset of the main dataset called 'sub' by oversampling the install=1 observation and (1/119) of the install=0 observations resulting in a sample with approximately equal number of events and non-events as shown below,

After Oversampling for Rare Events- Count of events '0' and '1'

Response counts in oversampled, subset data set

The FREQ Procedure

install	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1070	51.49	1070	51.49
1	1008	48.51	2078	100.00

- Following this, we perform the correction of the intercept as per the new oversampled dataset using the formula: $\beta_{corrected} = \beta_{estimated} + \log\left(\frac{p_1(1-y_1)}{y_1(1-p_1)}\right)$ to get the

Percent of Total Frequency	Percent of Total Frequency	p1	r1	w	off
0.8307304329	48.508180943	0.0083073043	0.4850818094	1.9259228239	4.7225876304

- In the next step, we run the logistic procedure on the oversampled dataset if the model remains unadjusted, i.e. how the intercept and co-efficient of the predictors change when the model is adjusted to handle the rare events but without performing the necessary corrections. The results are as follows,

Unadjusted Intercept

Analysis of Maximum Likelihood Estimates						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-2.6886	1.0322	6.7847	0.0092
device_volume		1	0.2032	0.1453	1.9548	0.1621
wifi		1	0.3747	0.1019	13.5132	0.0002
resolution		1	-1.2245	0.5990	4.1789	0.0409
device_height		1	0.00188	0.000776	5.8619	0.0155
device_width		1	0.00179	0.000773	5.3726	0.0205
publisher_id_class		1	-0.0665	0.0170	15.3111	<.0001
device_os_class		1	-0.0247	0.0156	2.5091	0.1132
device_make_class		1	0.0654	0.0205	10.2265	0.0014
device_platform_clas	android	1	-0.0462	0.1945	0.0563	0.8124

From the screenshot above, we observe that without the necessary corrections, the intercept differs significantly from the intercept of the original model. Hence, the unadjusted model should not be considered for the final model selection.

- In the next step, we run the oversampled dataset using the weight adjusted model which yields better results compared to the unadjusted model as shown below,

Adjusted Intercept- Weight adjusted model

Analysis of Maximum Likelihood Estimates						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-7.5213	5.8428	1.6571	0.1980
device_volume		1	0.1351	0.7747	0.0304	0.8616
wifi		1	0.3933	0.5753	0.4673	0.4942
resolution		1	-1.2739	3.2893	0.1500	0.6985
device_height		1	0.00190	0.00422	0.2032	0.6521
device_width		1	0.00186	0.00427	0.1894	0.6634
publisher_id_class		1	-0.0646	0.0907	0.5071	0.4764
device_os_class		1	-0.0242	0.0854	0.0801	0.7772
device_make_class		1	0.0669	0.1101	0.3692	0.5434
device_platform_clas	android	1	-0.1283	1.0862	0.0140	0.9060

- As a secondary approach, we also run the offset adjusted model which yielded the following results,

Adjusted Intercept- Offset adjusted model

Analysis of Maximum Likelihood Estimates						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-7.4113	1.0322	51.5558	<.0001
device_volume		1	0.2032	0.1453	1.9549	0.1621
wifi		1	0.3748	0.1019	13.5149	0.0002
resolution		1	-1.2246	0.5990	4.1795	0.0409
device_height		1	0.00188	0.000776	5.8627	0.0155
device_width		1	0.00179	0.000773	5.3734	0.0204
publisher_id_class		1	-0.0665	0.0170	15.3132	<.0001
device_os_class		1	-0.0247	0.0156	2.5096	0.1132
device_make_class		1	0.0654	0.0205	10.2279	0.0014
device_platform_clas	android	1	-0.0462	0.1945	0.0563	0.8124
off		0	1.0000	0	.	.

Comparing the weight-adjusted model and offset-adjusted model, we see that the weight-adjusted model's parameter coefficients are closer to the original dataset. Therefore, we decide to select the weight-adjusted model as our final model for handling the rare events.

ROC Curves:

Initial Linear Probability Model- ROC curve:

The first step to create the ROC curve is to build the model based on the training dataset. Then using the learnings from this model, the probability predictions are made on the unseen test data. And then finally using these predictions, the roc curve is drawn for the probability values of test data.

Since the ROC curve is drawn for the initial linear probability model, all the 10 predictors were used in the model. For Linear Probability Model the model generated with PROC REG step and then the model is used for scoring the test data. The predicted probabilities are then used in the PROC LOGISTIC to generate the ROC curve.

Initial Linear Probability Model- ROC curve

True Parameters: -8.1248 (intercept)
Offset-adjusted Model

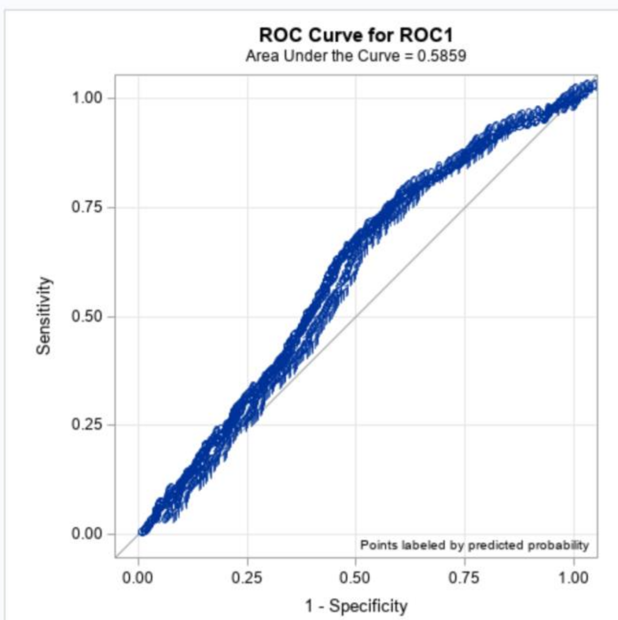
The LOGISTIC Procedure

Model Information	
Data Set	WORK.AD_LIN_PREDICT_INITIAL
Response Variable	install
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	36401
Number of Observations Used	36401

Response Profile		
Ordered Value	install	Total Frequency
1	0	36073
2	1	328

Probability modeled is install='1'.



ROC Association Statistics							
	Mann-Whitney						
ROC Model	Area	Standard Error	95% Wald Confidence Limits		Somers' D	Gamma	Tau-a
ROC1	0.5859	0.0144	0.5578	0.6141	0.1719	0.1719	0.00307

Final Linear Probability Model- ROC curve:

Similarly, the same procedure is repeated for the final models with the 8 predictors. The ROC plot is as below.

Final Linear Probability Model- ROC curve

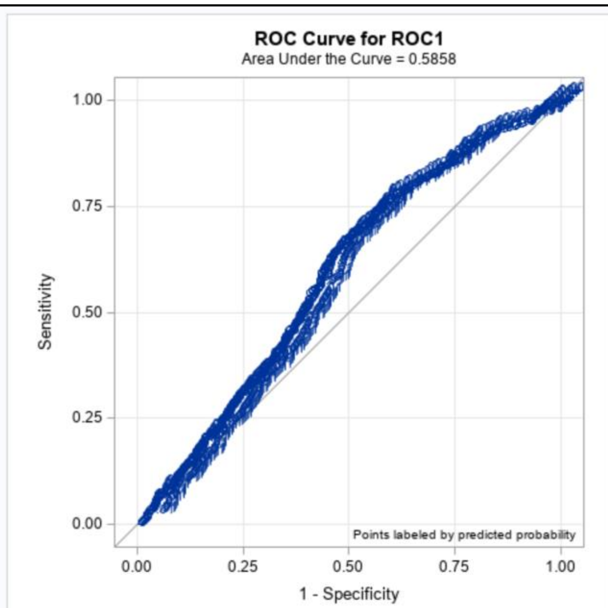
True Parameters: -8.1248 (intercept)
Offset-adjusted Model

The LOGISTIC Procedure

Model Information	
Data Set	WORK.AD_LIN_PREDICT_FINAL
Response Variable	install
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	36401
Number of Observations Used	36401

Response Profile		
Ordered Value	install	Total Frequency
1	0	36073
2	1	328



ROC Association Statistics							
	Mann-Whitney						
ROC Model	Area	Standard Error	95% Wald Confidence Limits		Somers' D	Gamma	Tau-a
ROC1	0.5858	0.0143	0.5576	0.6139	0.1715	0.1716	0.00306

From both these results we see that ROC curve looks similar and the area under the curve remains almost the same. However, it should be noted that the final model was able of reach the same Area under the curve value inspite of using less predictors.

Initial Logistic Regression Model- ROC curve:

Since the ROC curve is drawn for the initial Logistic Regression model, all the 10 predictors were used in the model. For Logistic Regression Model the model is first generated with PROC LOGISTIC step and then the model is used for scoring the test data. The predicted probabilities from the test data are then used in the PROC LOGISTIC statement again and the ROC curves are generated with the plots=roc(id=prob) and roc pred=p_1 statement. The results and ROC curve for the initial Logistic probability model is as below.

Initial Logistic Regression Model- ROC curve

True Parameters: -8.1248 (intercept)
Offset-adjusted Model

The LOGISTIC Procedure

Model Information		
Data Set	WORK.AD_LOGIT_PREDICT_INITIAL	Posterior Probabilities for DATA=WORK.AD_TEST_WITH_INDICATORS.
Response Variable	install	
Number of Response Levels	2	
Model	binary logit	
Optimization Technique	Fisher's scoring	

Number of Observations Read	36401
Number of Observations Used	36401

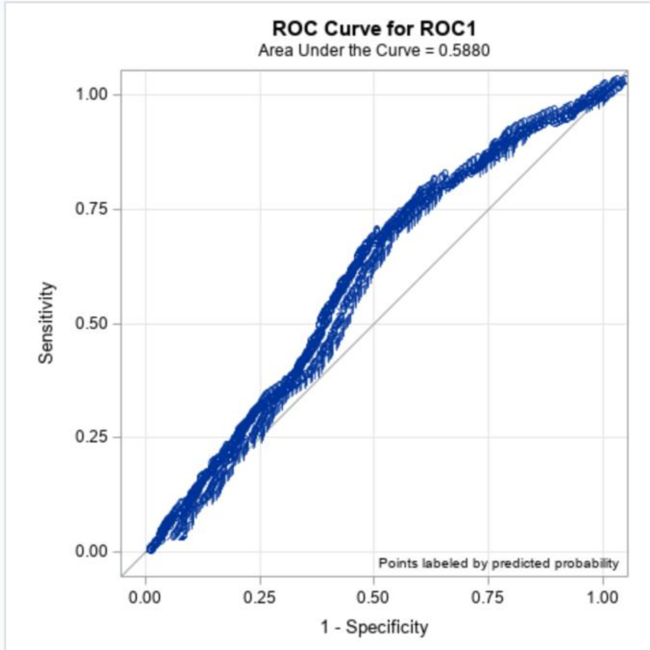
Response Profile		
Ordered Value	install	Total Frequency
1	0	36073
2	1	328

Probability modeled is install='1'.

Score Test for Global Null Hypothesis		
Chi-Square	DF	Pr > ChiSq
33.9388	9	<.0001

ROC Model: ROC1

ROC Model Information		
ROC Contrast Coefficients	P_1	Predicted Probability: install=1



ROC Association Statistics							
	Mann-Whitney						
ROC Model	Area	Standard Error	95% Wald Confidence Limits		Somers' D	Gamma	Tau-a
ROC1	0.5880	0.0144	0.5598	0.6161	0.1759	0.1760	0.00314

Final Logistic Regression Model- ROC curve:

Following the same procedure as done in Initial logistic model, the ROC curve for the final model is also generated with the corresponding predictors that were chosen for the final model. The results and ROC curve for the final Logistic probability model is as below.

Final Logistic Regression Model- ROC curve

True Parameters: -8.1248 (intercept)
Offset-adjusted Model

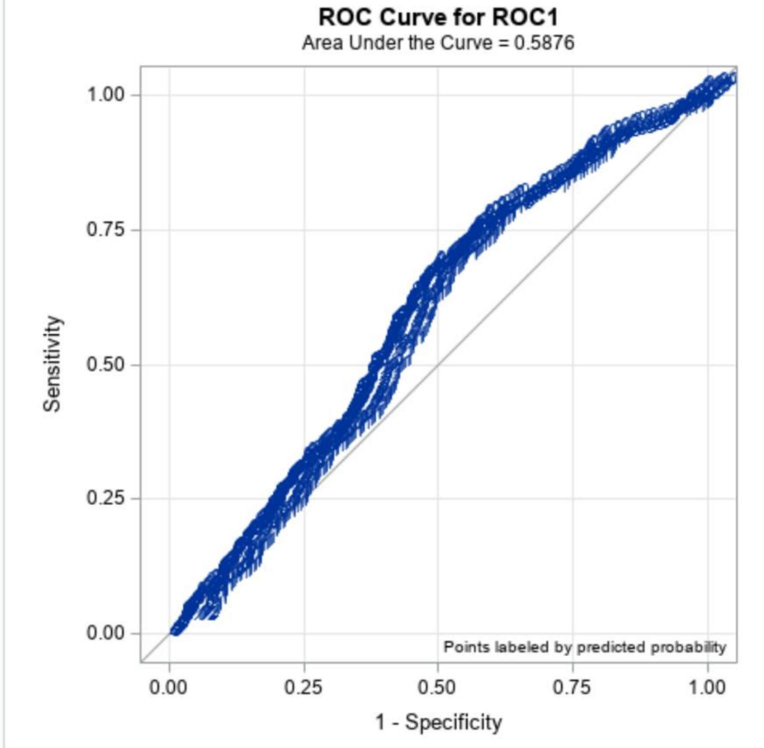
The LOGISTIC Procedure

Model Information		
Data Set	WORK.AD_LOGIT_PREDICT_FINAL	Posterior Probabilities for DATA=WORK.AD_TEST_WITH_INDICATORS.
Response Variable	install	
Number of Response Levels	2	
Model	binary logit	
Optimization Technique	Fisher's scoring	

Number of Observations Read	36401
Number of Observations Used	36401

Response Profile		
Ordered Value	install	Total Frequency
1	0	36073
2	1	328

Probability modeled is install='1'.



ROC Association Statistics

ROC Model	Mann-Whitney Area	Mann-Whitney Standard Error	Mann-Whitney 95% Wald Confidence Limits		Somers' D	Gamma	Tau-a
ROC1	0.5876	0.0144	0.5595	0.6158	0.1753	0.1753	0.00313

Here, the initial and the final logistic models both have ROC curve that looks similar and the area under the curve remains the same. However, it should be noted that the final logistic model was able to reach the same Area under the curve value inspite of using less predictors which was 8.

AUC (area under the curve):

On further comparing the AUC values of various models we see that the logistic model has better values based on the results from the table below. Also, since the final linear probability model and the final logistic model has almost the same AUC value when compared with their corresponding Initial models inspite of using lower number of predictors, there are better models than initial models. Thus, only the final linear probability model and Final Logistic Regression Model is taken for consideration here.

SNO.	Model	Area Under the Curve	95% Confidence Interval	
			Lower Limit	Upper Limit
1	Final Linear Probability Model	0.5858	0.5576	0.6139
2	Final Logistic Regression Model	0.5876	0.5595	0.6158

Based on the values from the comparison table we see that, the Final logistic model has the highest AUC value of 0.5876 and also at 95% confidence level this same model has the higher value when compared to the Final probability model.

Part II

In this section of the project, we look to determine whether the advertising platform would like to show the ad from the advertiser depending on the different publisher and consumer characteristics.

Particularly, the objective is to decide on a threshold based on the ROC table such that if the probability of installing the ad is above that threshold, the ad is shown to the consumer.

In order to determine this threshold, firstly, we need to calculate the total expected cost for every choice occasion. The formula to calculate the total expected cost is as follows,

Total expected cost = # False positives*False positive cost + # False negatives*False negative cost

From the given information, we observe that there are two possible scenarios of the advertising platform incurring a loss as follows,

i. The first situation is where the platform shows an ad to a consumer who would not install the app. This results in some inconvenience to the consumer leading to less participation. This situation is identified as a false positive where the 'actual choice' is a consumer not installing the app whereas the 'predicted choice' is a consumer installing the app. The misclassification cost here is 1 cent ($0.01)

ii. The second situation is where the platform fails to show an ad to a consumer who would actually install the app. This results in a missed opportunity for the platform. This situation is identified as a false negative where the 'actual choice' is a consumer installing the app whereas the 'predicted choice' is a consumer not installing the app. The misclassification cost here is $1.

Logistic Regression Models -

We start by using the options available in proc logistic to generate the ROC tables automatically for both the initial and final logistic regression models. This is followed by creating a new column named 'total cost' which is calculated as # False positives * 0.01 + # False negatives * 1.

ROC table for initial logistic regression model

	Probability Level	No. of Correctly Predicted Events	No. of Correctly Predicted Nonevents	No. of Nonevents Predicted as Events	No. of Events Predicted as Nonevents	Sensitivity	1 - Specificity	total_cost1
1	0.0210951462	0	36072	1	328	0	0.0000277216	328.01
2	0.0207198693	0	36071	2	328	0	0.0000554431	328.02
3	0.0206603662	0	36070	3	328	0	0.0000831647	328.03
4	0.0204316434	0	36069	4	328	0	0.0001108863	328.04
5	0.0202322619	0	36068	5	328	0	0.0001386078	328.05
6	0.0202277645	0	36067	6	328	0	0.0001663294	328.06
7	0.0201741302	0	36066	7	328	0	0.000194051	328.07
8	0.0200787034	0	36065	8	328	0	0.0002217725	328.08
9	0.0200654553	0	36064	9	328	0	0.0002494941	328.09
10	0.0199744996	0	36063	10	328	0	0.0002772156	328.1

The minimum cost here was identified to be \$281.50. Through a simple query, the corresponding probability threshold was observed to be 0.00753.

Repeating the same procedure for the final logistic regression model, we get the following results,

ROC table for final logistic regression model

	Probability Level	No. of Correctly Predicted Events	No. of Correctly Predicted Nonevents	No. of Nonevents Predicted as Events	No. of Events Predicted as Nonevents	Sensitivity	1 - Specificity	total_cost2
1	0.0208458798	0	36072	1	328	0	0.0000277216	328.01
2	0.0205825925	0	36071	2	328	0	0.0000554431	328.02
3	0.020382632	0	36070	3	328	0	0.0000831647	328.03
4	0.020100982	0	36069	4	328	0	0.0001108863	328.04
5	0.0200983281	0	36068	5	328	0	0.0001386078	328.05
6	0.0199964371	0	36067	6	328	0	0.0001663294	328.06
7	0.0199293937	0	36066	7	328	0	0.000194051	328.07
8	0.0199056051	0	36065	8	328	0	0.0002217725	328.08
9	0.0198824396	0	36064	9	328	0	0.0002494941	328.09
10	0.0198275425	0	36063	10	328	0	0.0002772156	328.1

Through simple queries, the minimum cost was \$282.11 and the corresponding probability threshold was observed to be 0.00754.

Linear Probability Models -

In the case of our linear probability models however, we are required to manually generate the ROC table. This is done by the following procedure,

- Creating a new dataset for each probability threshold using the dataset previously created in the code which holds the probabilities of prediction and setting the selection indicator to 0 to generate the predictions only for the test data
- Apart from this, we impose a condition such that if the predictions generated > the probability threshold, then 'predicted choice' =1.
- We then create two new columns -
  'false positive' as if install=0 and predicted=1 then false_pos=1; and
  'false negative' as if install=1 and predicted=0 then false_neg=1;

The sample table with false positive and false negative values created based on "Predicted Choice" value is as given below:

Final linear probability model with False Positive and False Negative values

	install	Selection Indicator	Predicted Value of install	predicted	false_pos	false_neg
1	0	0	0.0083499377	1	1	.
2	0	0	0.0073573979	1	1	.
3	0	0	0.0056883155	1	1	.
4	0	0	0.0056168161	1	1	.
5	0	0	0.0087344199	1	1	.
6	0	0	0.0053624524	1	1	.
7	0	0	0.0059336728	1	1	.
8	0	0	0.0094962068	1	1	.
9	0	0	0.0101313011	1	1	.
10	0	0	0.0047855553	1	1	.

- Once we create the datasets, we create a table for each of these datasets to generate the count of false positives and false negatives at each probability threshold

Final linear probability model with False Positive and False Negative values for probability = 0.001

	install	Selection Indicator	Predicted Value of install	predicted	false_pos	false_neg	count_fp	count_fn
1	0	0	0.0083499377	1	1	.	36067	0
2	0	0	0.0073573979	1	1	.	36067	0
3	0	0	0.0056883155	1	1	.	36067	0
4	0	0	0.0056168161	1	1	.	36067	0
5	0	0	0.0087344199	1	1	.	36067	0
6	0	0	0.0053624524	1	1	.	36067	0
7	0	0	0.0059336728	1	1	.	36067	0
8	0	0	0.0094962068	1	1	.	36067	0
9	0	0	0.0101313011	1	1	.	36067	0
10	0	0	0.0047855553	1	1	.	36067	0

- Next, we create a table by manually inputting these counts at the respective thresholds to summarize the ROC table for both the initial and final linear probability models.
- Finally, we calculate the total cost as per the formula used before at each threshold to observe the lowest total cost

ROC table for initial linear probability model

	probability	false_positive	false_negative	total_cost
1	0.001	36061	0	360.61
2	0.005	32438	0	324.38
3	0.01	7611	0	76.11
4	0.015	125	0	1.25
5	0.02	0	0	0
6	0.025	0	0	0
7	0.03	0	0	0
8	0.035	0	0	0
9	0.04	0	0	0
10	0.045	0	0	0
11	0.05	0	0	0

ROC table for final linear probability model

	probability	false_positive	false_negative	total_cost
1	0.001	36067	0	360.67
2	0.005	32438	0	324.38
3	0.01	7611	0	76.11
4	0.015	125	0	1.25
5	0.02	0	0	0
6	0.025	0	0	0
7	0.03	0	0	0
8	0.035	0	0	0
9	0.04	0	0	0
10	0.045	0	0	0
11	0.05	0	0	0

Summarizing the ROC tables for all the four models, we see that the tables for the logistic regression models do not have all the set probability thresholds as the linear probability models.

For proper purposes of comparison, since the lowest costs for the logistic models occur at the thresholds ~0.007, we compare these costs with the costs at the probability threshold of 0.005 for the linear models as follows,

Model	Probability Threshold	Minimum total cost
Initial logistic regression model	0.00753	$281.50
Initial linear probability model	0.005	$324.38
Final logistic regression model	0.00754	$282.11
Final linear probability model	0.005	$324.38

From the table above, though there are not much difference between the cost of initial and final logistic model we can conclude that the initial logistic regression model provides the lowest total cost at probability 0.007.