**BUAN 6337 Predictive Analytics using SAS**

**Project 2**

**Predictive Models for Mobile Advertising**

The data for this project comes from the mobile advertising space. In order to encourage consumers to install its app (e.g. a game), an app *developer* advertises its app on other apps (e.g., other games) through a mobile *advertising platform*. These other apps are developed by other game *publishers*. Consumers who view the ads on these other apps, can click on the ad to install the app from the developer. We will refer to the advertising app developer as the *advertiser* and the other apps as *publishers*. See figure below.

App Developer / Advertiser

Advertising Platform

.

.

.

Publisher 1

Publisher 2

Publisher k-1

Publisher k

Consumer

Consumer

Consumer

Consumer

.

.

.

Install

Not Install

The dataset for this project contains data about ads from one particular advertiser through multiple publishers. 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. The description of the variables are given below.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Type** | **Description** |
| publisher\_id\_class | Categorical | Publisher Id |
| device\_make\_class | Categorical | Device Manufacturer |
| device\_platform\_class | Categorical | Phone OS Type (iPhone / Android) |
| device\_os\_class | Categorical | Phone OS Version |
| 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 | Numerical | Whether WiFi was enabled when ad was displayed (Yes = 1, No = 0) |
| Install | Binary | Whether Consumer Installed Advertiser’s App (Yes = 1, No = 0) |

**Part I.**

The advertiser needs to determine how much to pay for placing an ad, depending on the publisher and on the consumer characteristics. The optimal payment is proportional to the probability that a consumer seeing the ad will install the ad.

1. Develop a **linear probability model** to predict the probability of installing the ad based on publisher and consumer characteristics. Describe in detail your approach for model building, evaluation and selection. Present your final model and performance metrics.

The description of your approach should include, for example, what variables to include in your model building process (and why), did you create new variables from existing ones (and why), how / what alternative models did you consider, how did you compare these alternative models and why did you compare these models in this way.

1. Develop a **logistic regression model** to estimate the probability of installing the ad based on publisher and consumer characteristics. Describe your approach as in part (a) above – elaborating only what is new or different than above. Present your final model and performance metrics.

In particular, discuss whether you need to consider modeling of rare events in this case – why / why not? Compare the results with and without considering rare events - (i) estimate the model without considering rare events, and (ii) estimate the model using oversampling approach for handling rare events and then applying the correction to obtain the corrected intercept

(Note: See lecture for how you can to calculate the correction after estimating the model. One approach is to implement this correction using a DATA step after estimating the model with the oversampling approach. Another approach is to directly implement through PROC LOGISTIC - see [support.sas.com/kb/22/601.html](http://support.sas.com/kb/22/601.html) for how to do this).

**Part II**

The advertising platform would like to determine whether to show the ad from this advertiser 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.

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 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 to the platform. The platform would like to minimize the total expected cost.

1. For each of the above models you estimated in part I above, generate the ROC table using SAS, and plot the total cost for different threshold values. (question contd. next page)

Note that for the linear probability model (unlike the logistic regression model), SAS does not generate the ROC table automatically. You will need to write a proc or data step to create the table yourself.

To make your job easier, you can calculate the total cost at these thresholds:

0.001 0.005 0.010 0.015 0.020 0.025 0.030 0.035 0.040 0.045 0.050

1. Which of these models provide the lowest total cost?

(For the logistic regression model for rare outcomes, you cannot use the oversampled data to calculate the cost since this is not representative of the actual distribution of outcomes.)

Deliverables

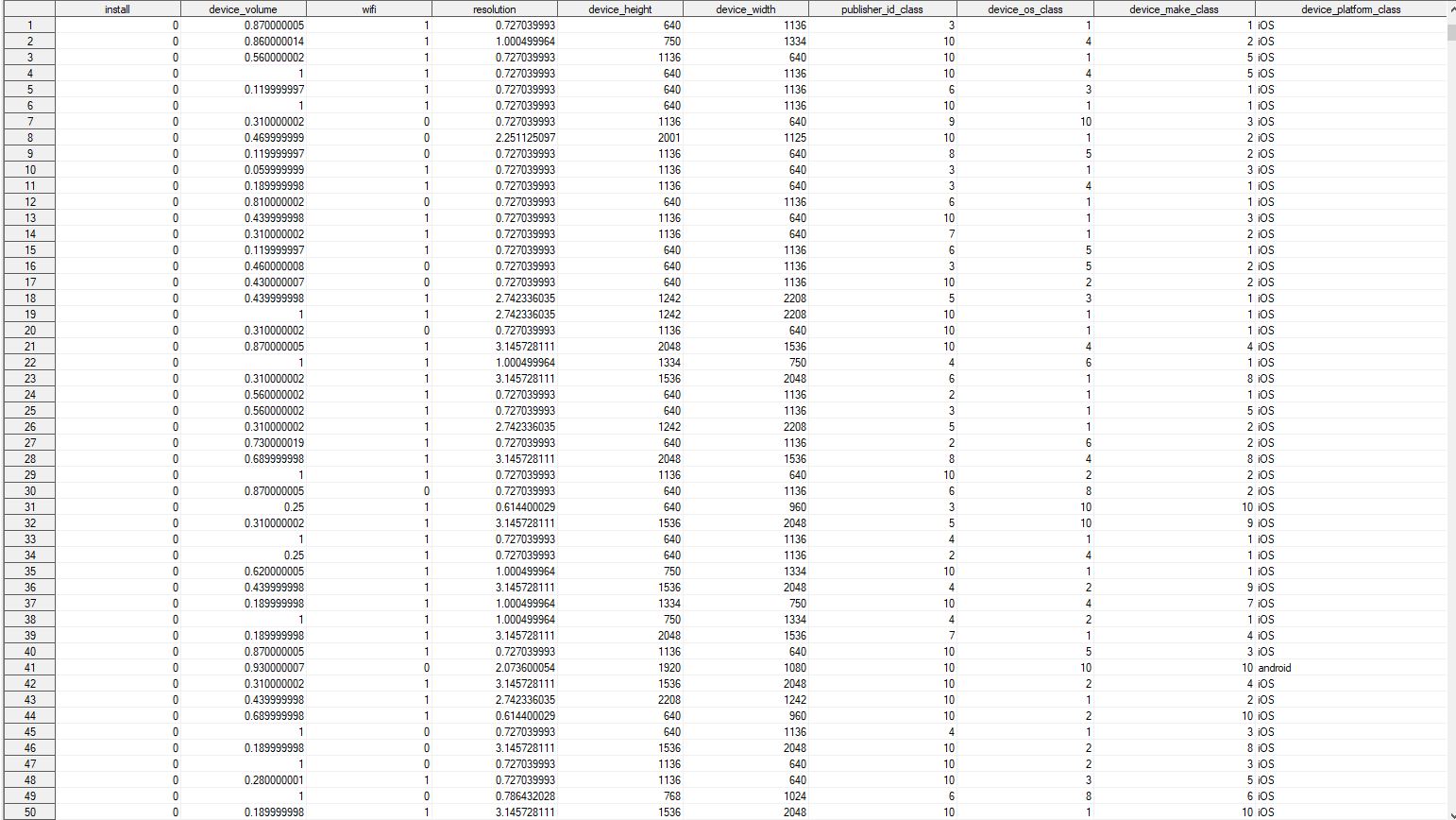
* Project Report: For each question above, describe the model building and selection process that you followed, along with suitable tables and graphs as necessary. Upload 1 pdf/word file for the entire project which includes your description for all the questions.
* SAS code: Include a SAS file with detailed comments to reproduce all the results, tables and figures in the report. The code must be clearly labeled so that it is straightforward to see how to reproduce a particular result / table / figure. Make sure your codes can be executed properly when uploaded, as it is part of your project score.

**Approach:**

Part One

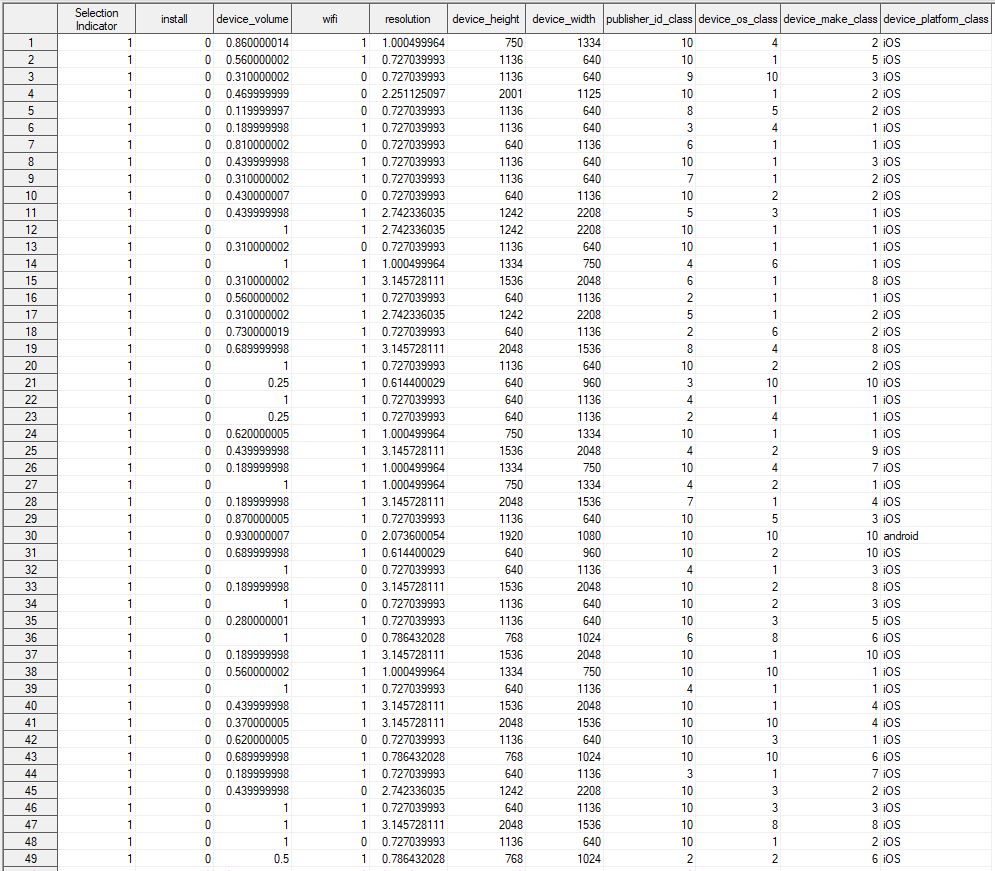
* **Linear probability models**

First, set the working directory using Libname and import the dataset ‘ADS.DATA‘ using DATA.

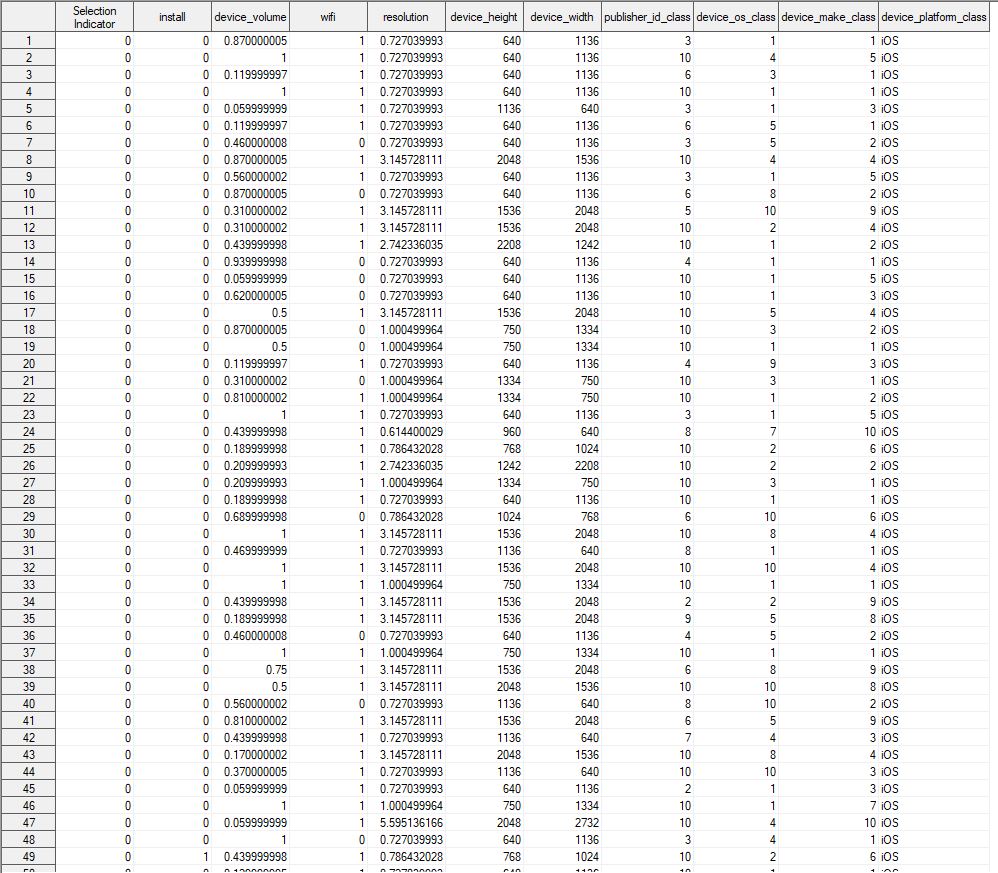
The dataset: 

Next we will split the dataset into Training and Test Dataset sample. We will split the dataset in the ratio 80% Training and 20% Test dataset

Training dataset:

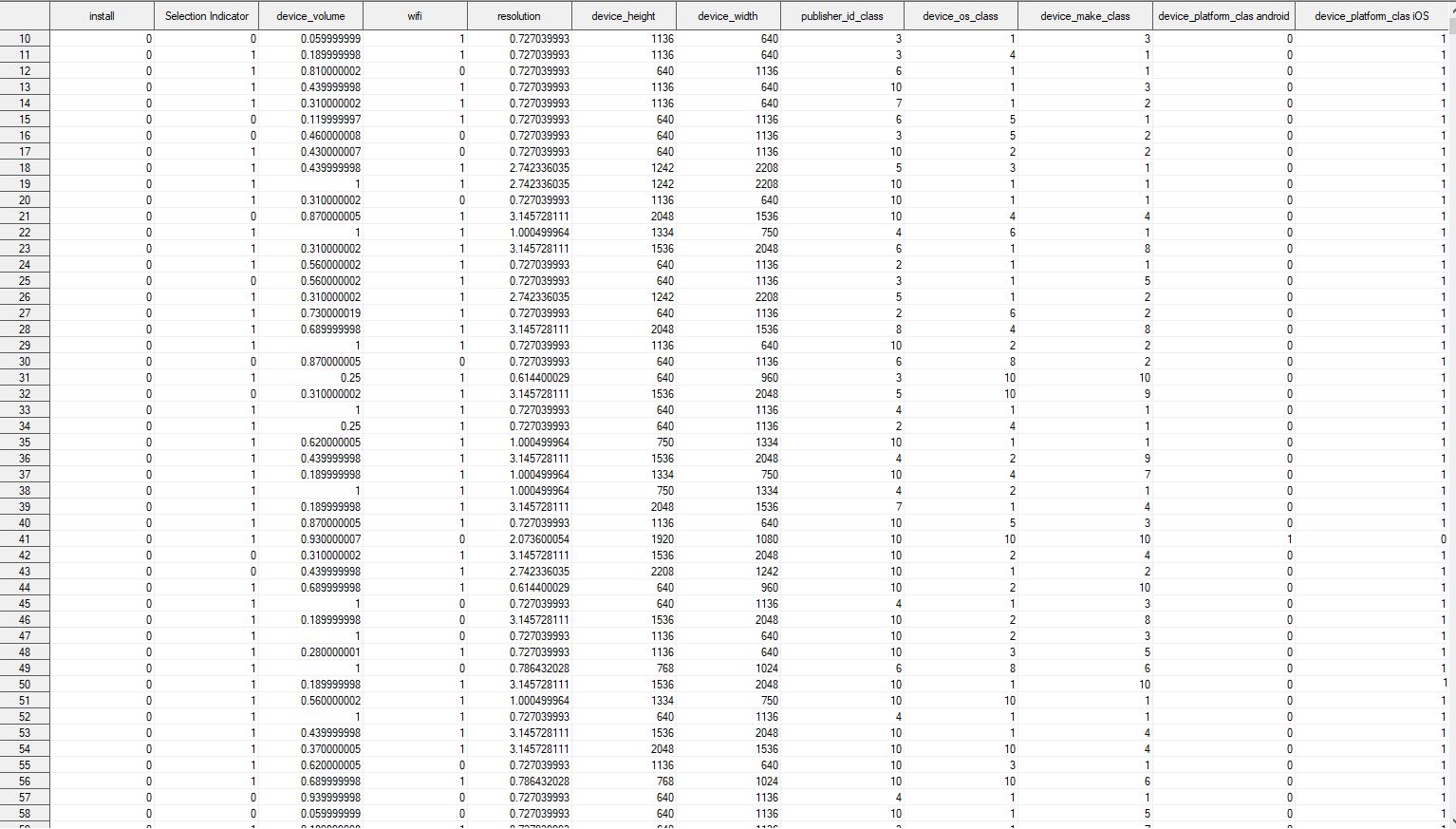


Testing Dataset:

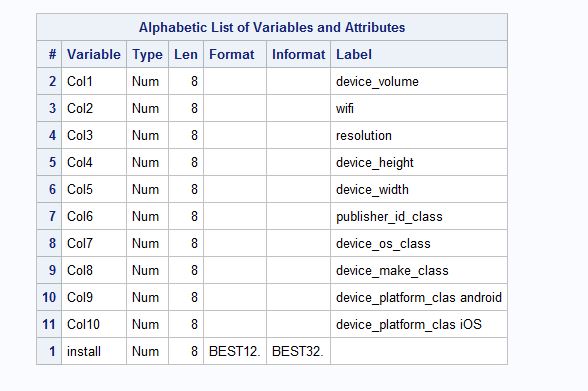


As the dataset have categorical variables, so converting it into numerical variables using the proc glmmod.

The encoding for the initial dataset. Notice the device\_platform\_class\_ prefixed columns:

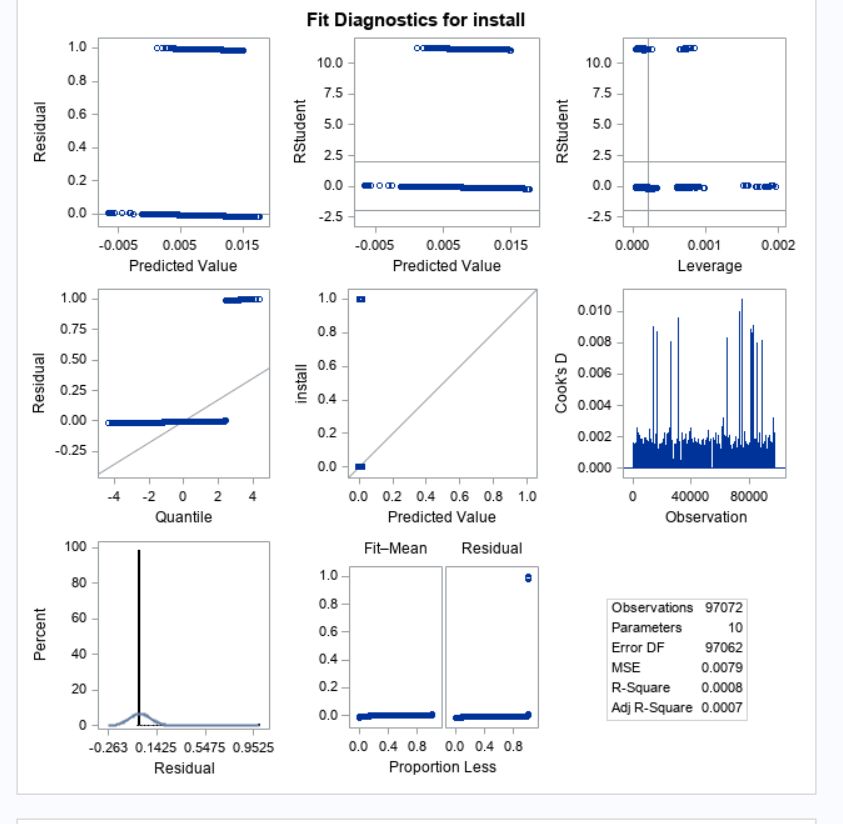


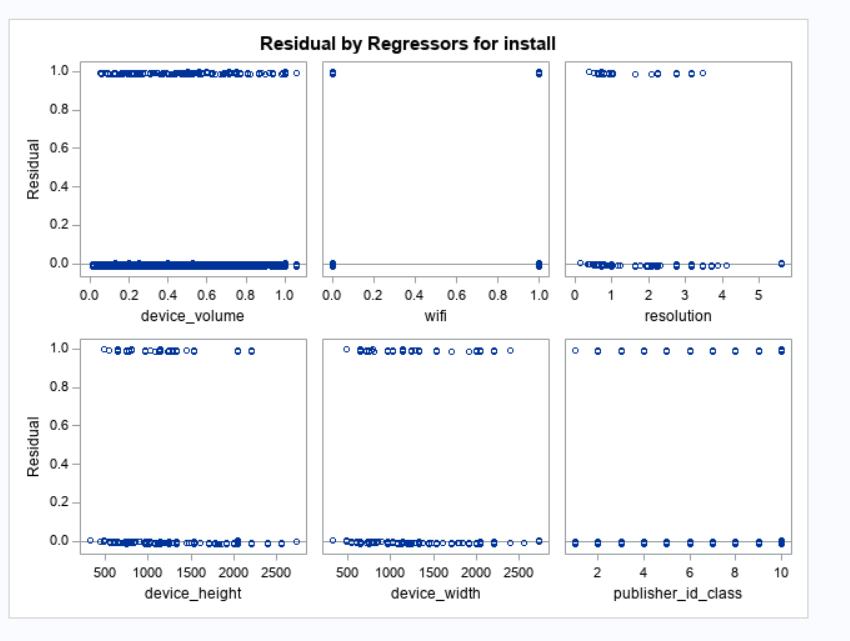
Using the proc contents statement to provide alias names as column numbers to the feature variables.

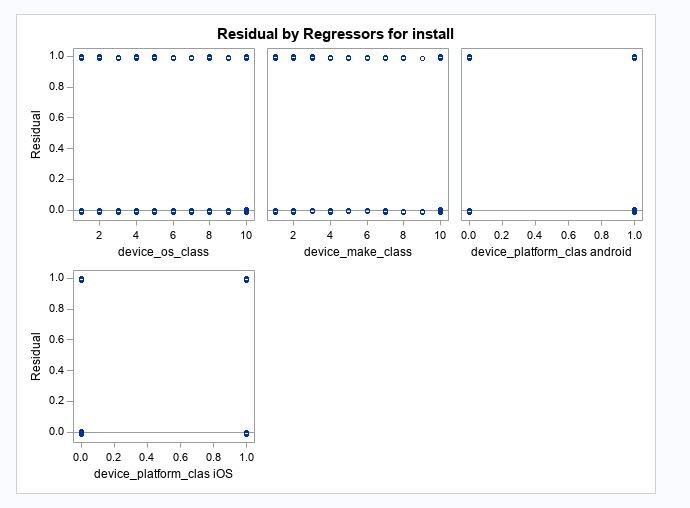


1. Linear Probability Model:

Ran a linear model (Kitchen sink regression) taking install as dependent and rest as independent variable using the Proc reg.

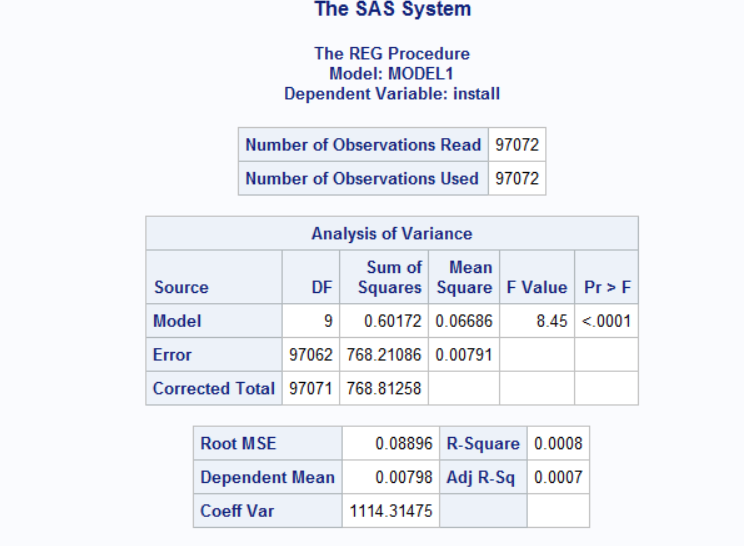


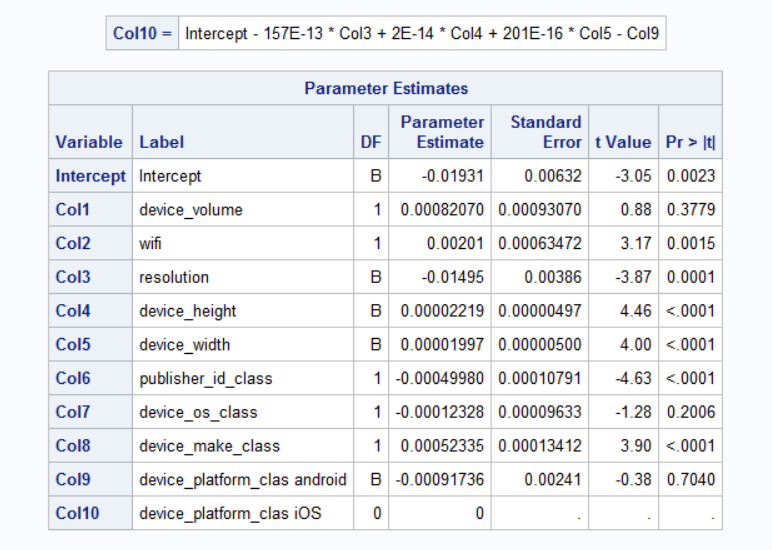




As we can observe from the Fit Diagnostic, there are lot of outliers in the dataset which can result into poor model. So we imputed those values to make sure the variance effect of the outliers does not affect the model.

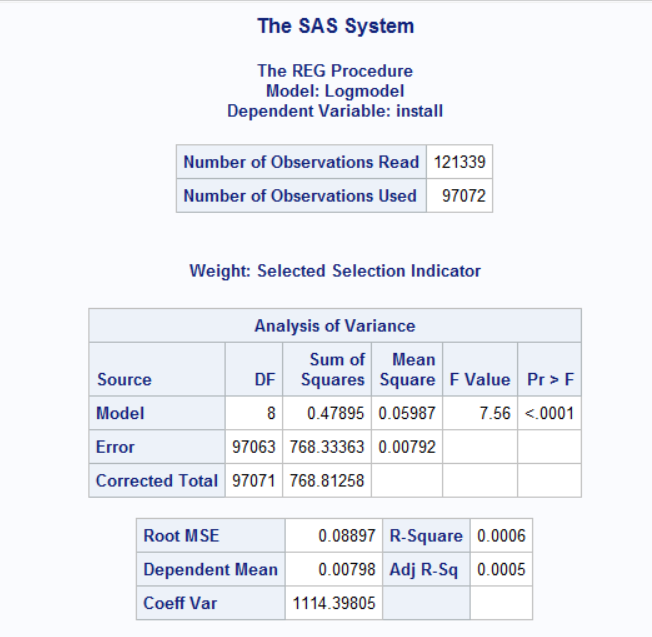
Initial Linear Model :

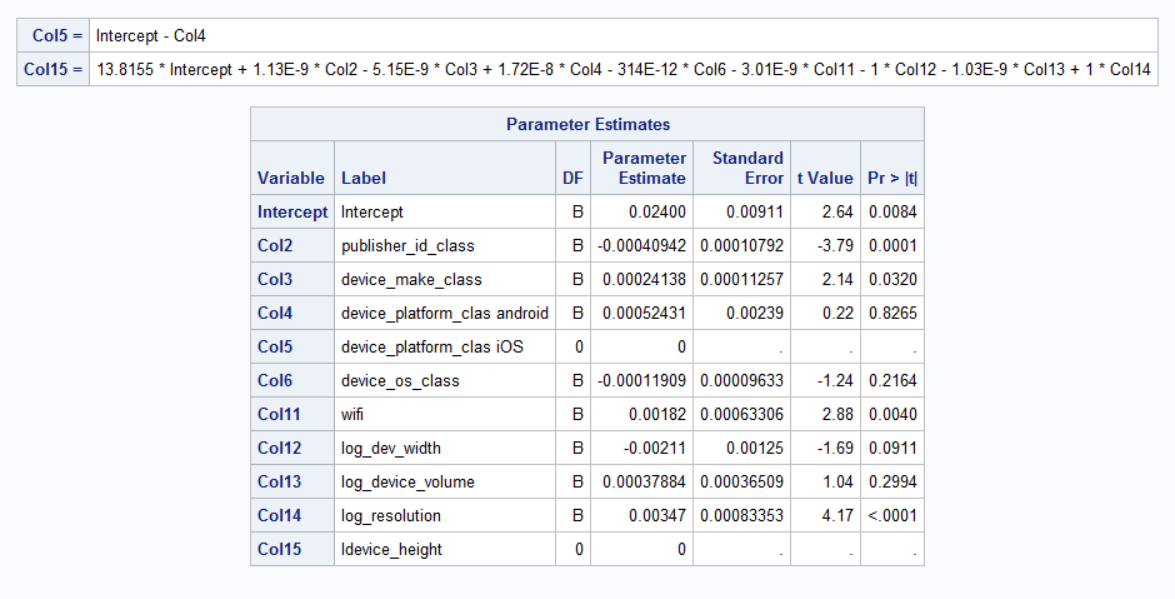




Here we can observe that all the variables have significant p – values at 1%, 5% and 10% levels.

Now we will run a log model to check if the significance of the variables becomes more significant.

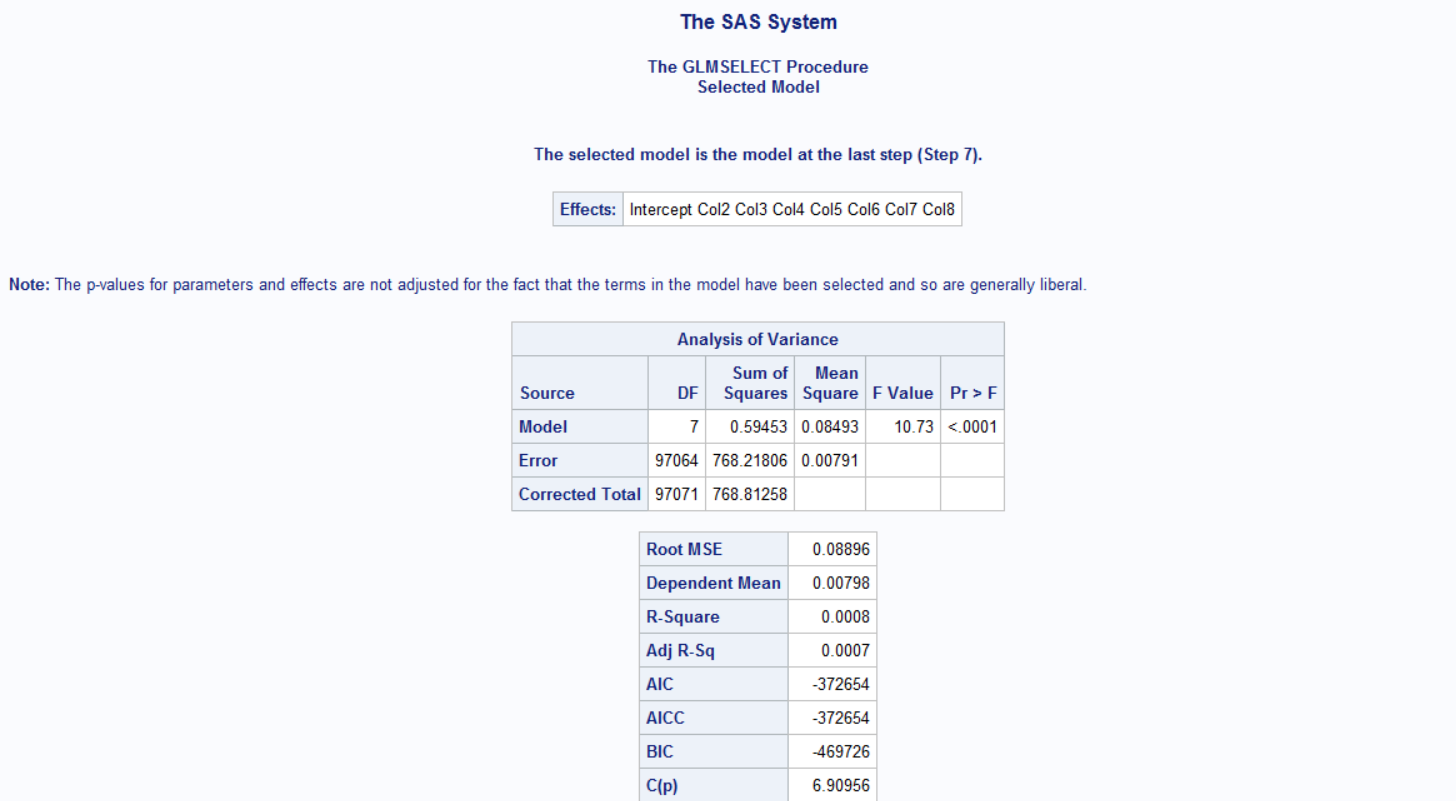


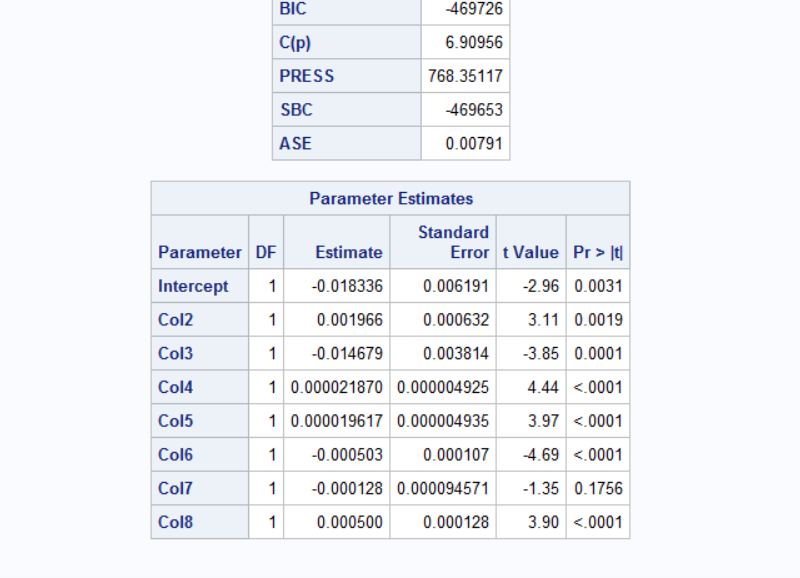


We can see that linear model has more significant values than the log model so we will move forward with Linear model.

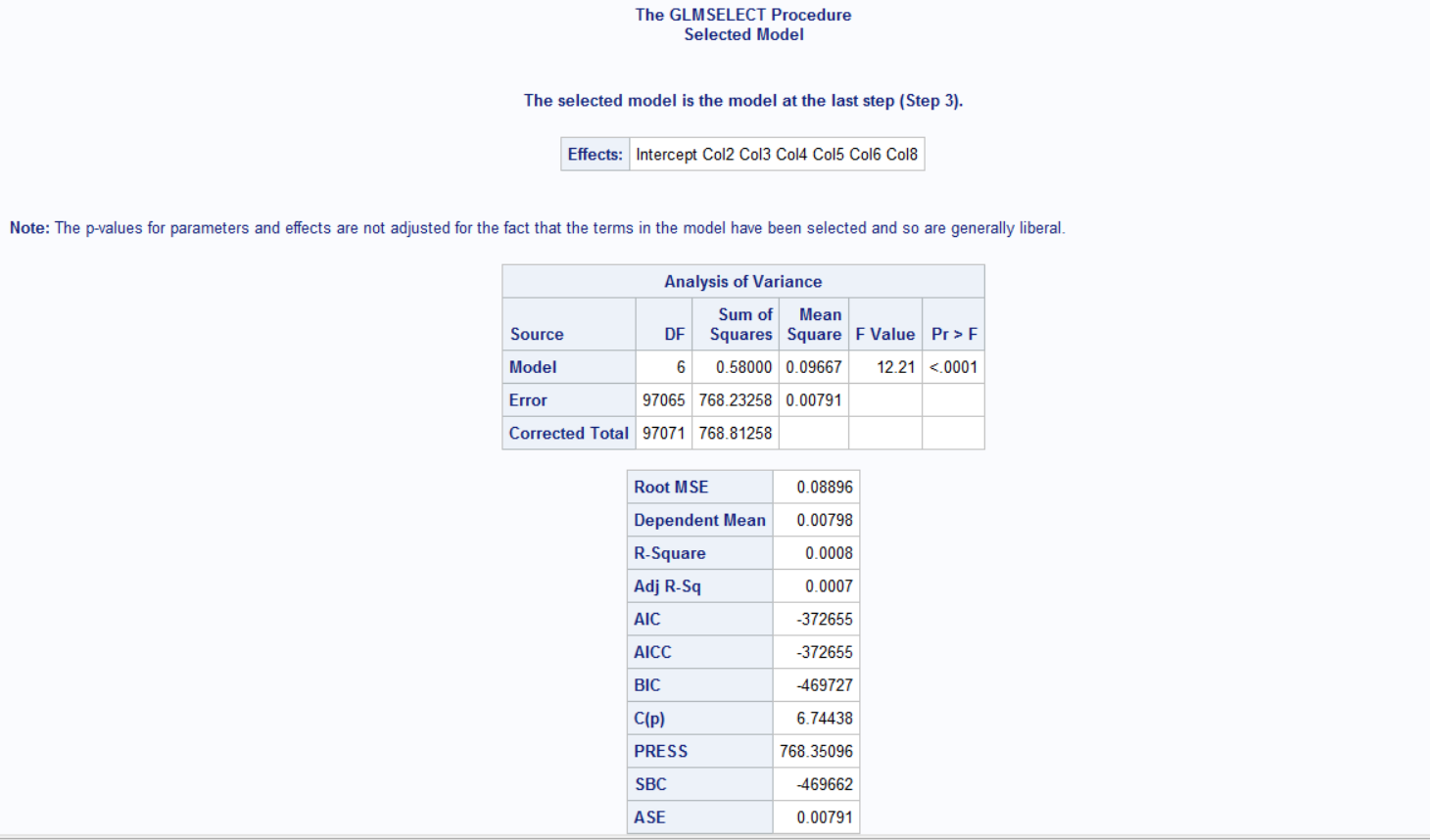
Next, selecting variables which are suitable for the model and removing other variables using various methods.

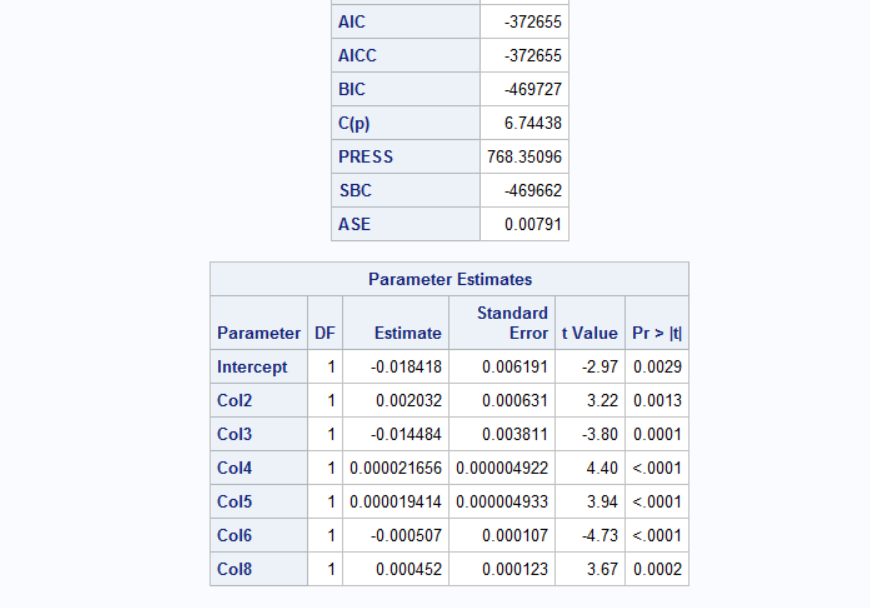
Forward Selection:



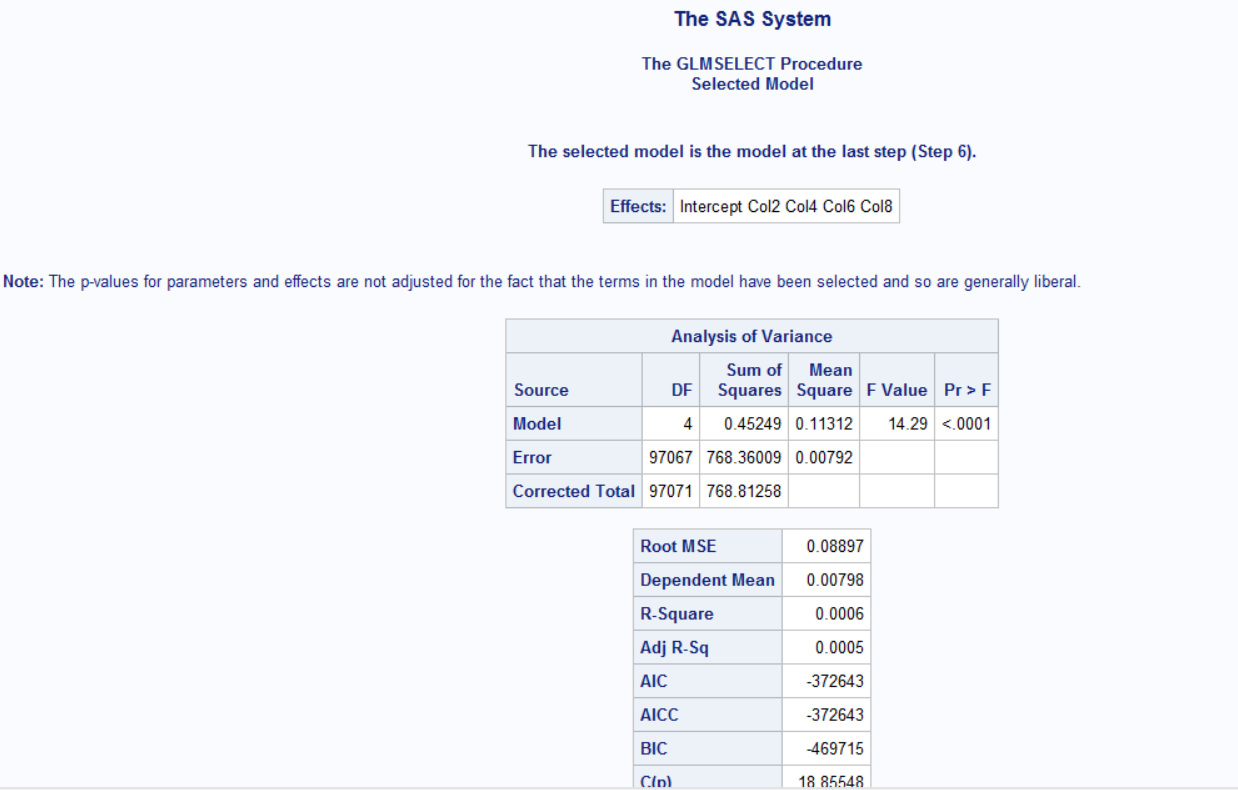


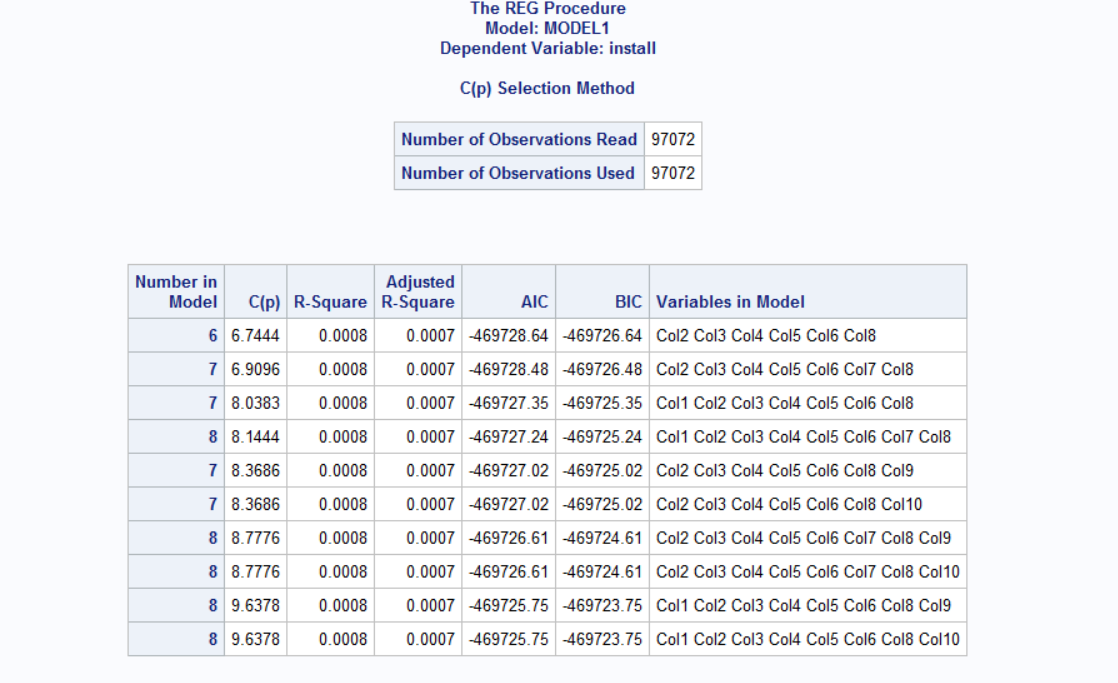
Backward Selection:





Stepwise:

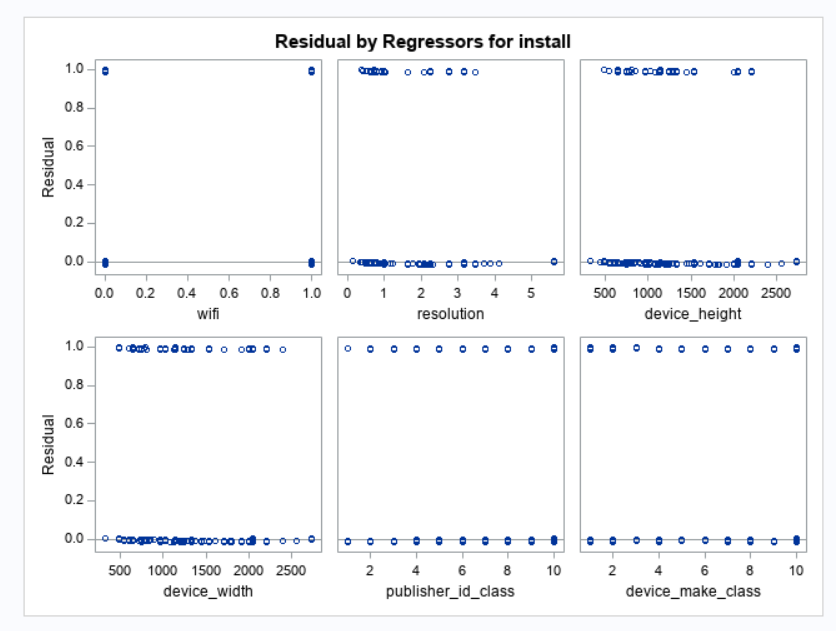
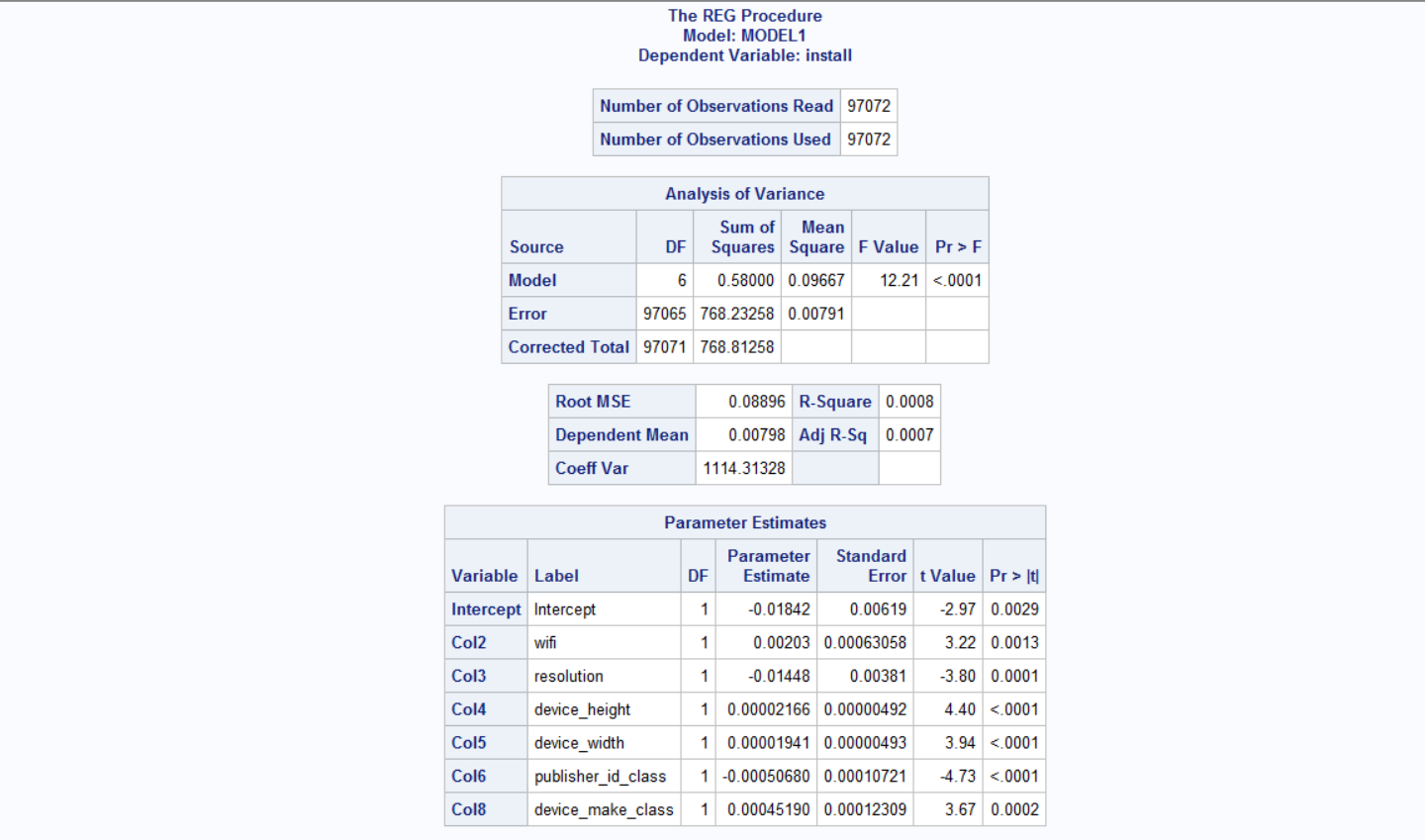


Best Models:

From the above methods the best predictors and the final model which involves the following columns:

* + Col 2
  + Col 3
  + Col 4
  + Col 5
  + Col 6
  + Col 8

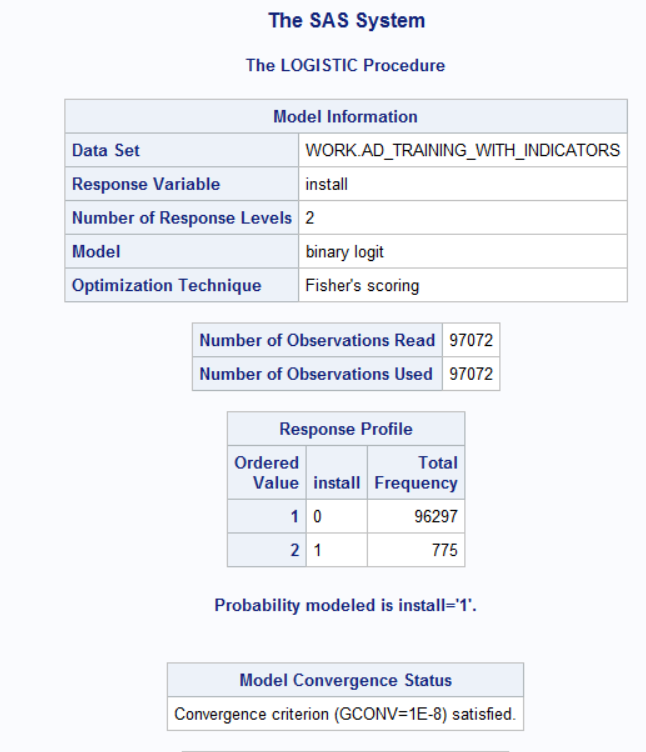
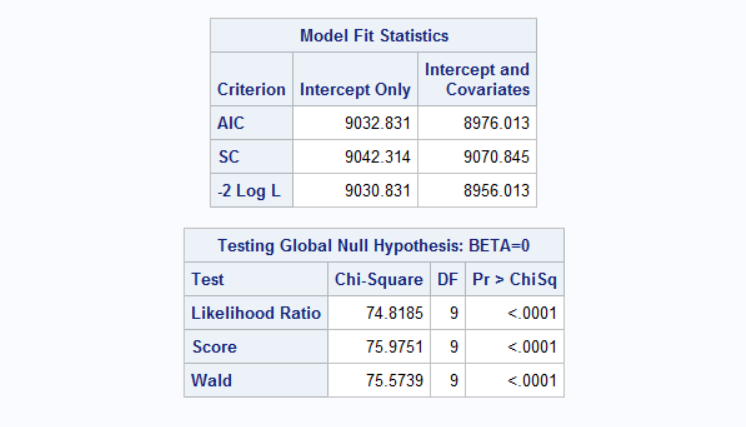
Finale Linear Model:

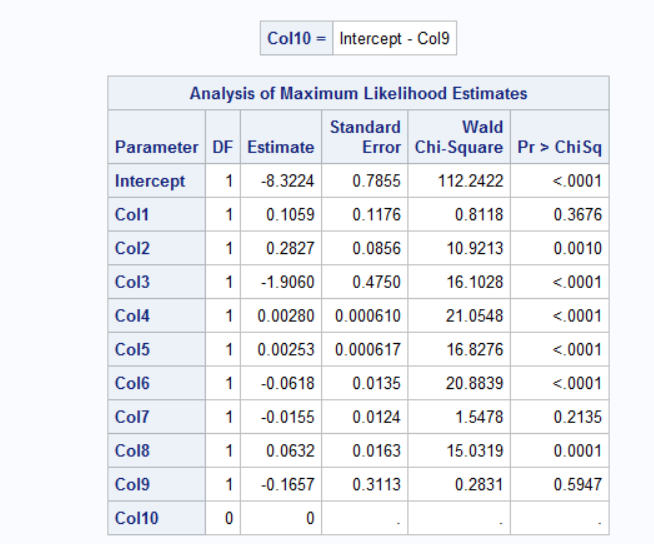
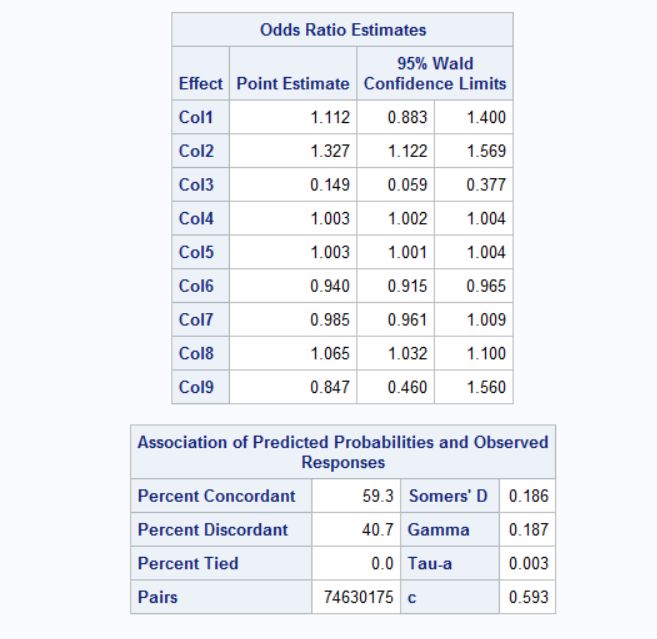


* + Although the predicted variables are within probability range, the residuals have to normal which has to be proved using plots.
  + The std error estimates will be invalid as the normality assumption is violated and therfore the hypothesis testing on predictor variables wouldn’t be valid.

1. Logistic Model

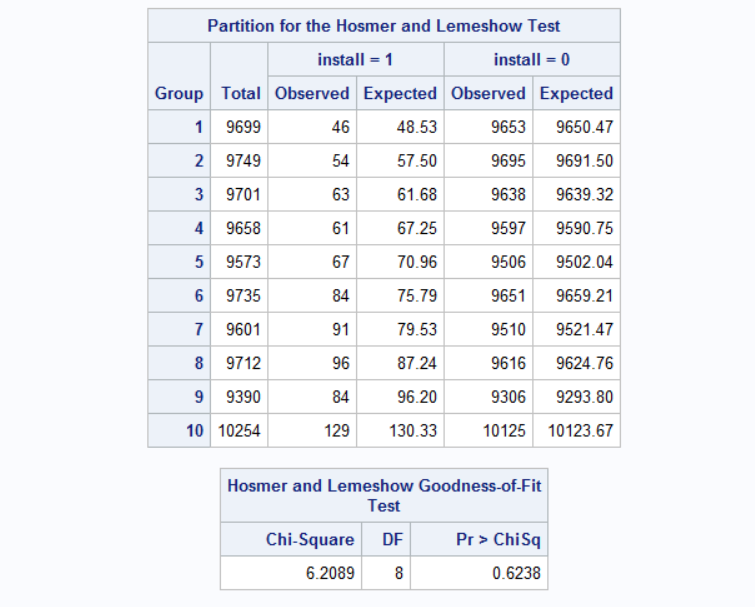
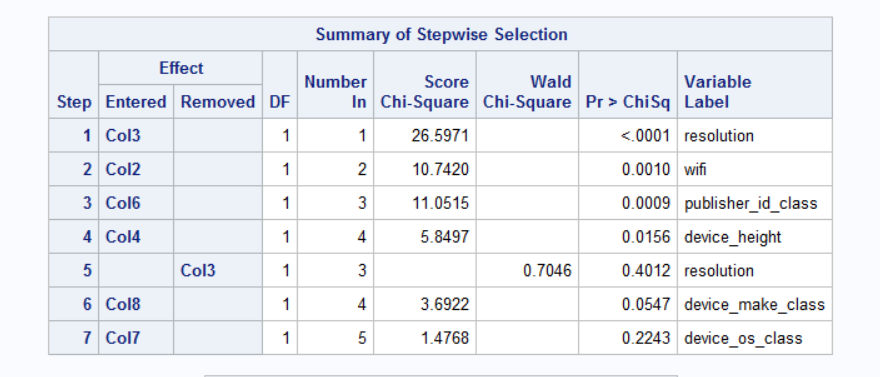
Initial Logistic Model:

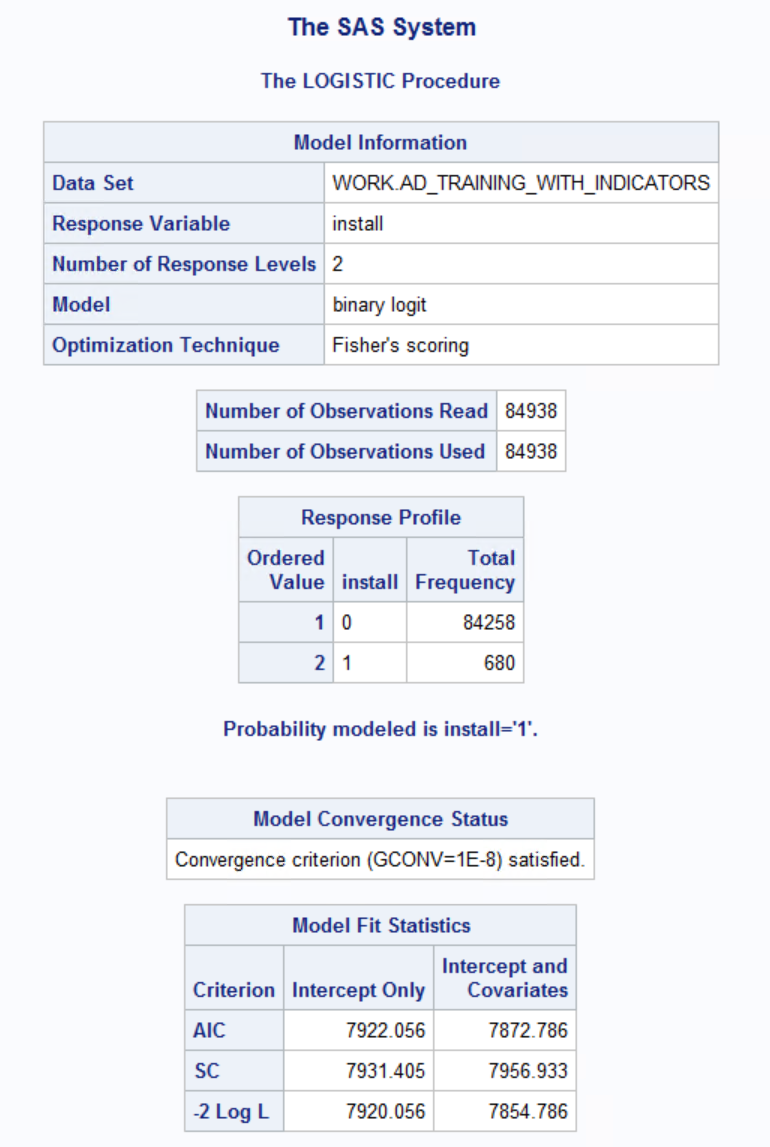
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trial | Selection Method | Selection Criteria & Parameters | Number of Predictors | Log-likelihood |
| 1 | Stepwise Selection | Significance level: -  Entry value: 0.25  Stay value: 0.35 | 8 | 8973.616 |
| 2 | Forward Selection | Significance level: -  Entry value: 0.25 | 8 | 8957.097 |
| 3 | Backward Selection | Significance level: -  Stay value: 0.35 | 8 | 8956.306 |

Stepwise:



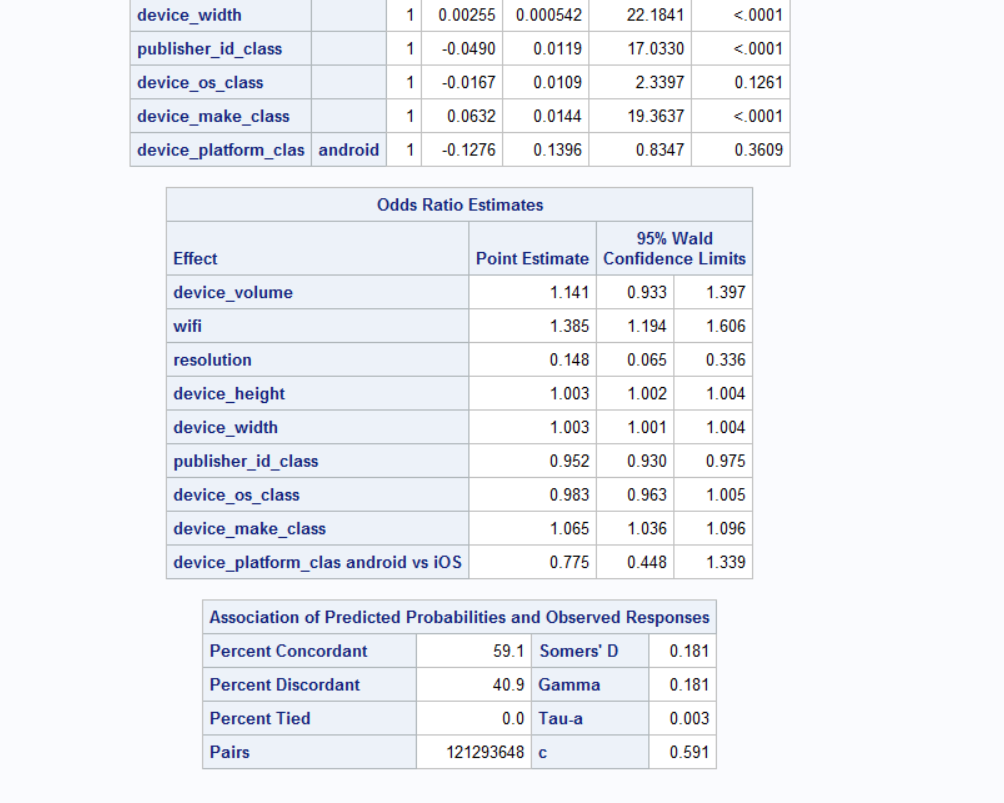
Final Logistic Model:

(i)Estimation of the model without considering rare event









Most of the variables are significant to the 0.01% level.

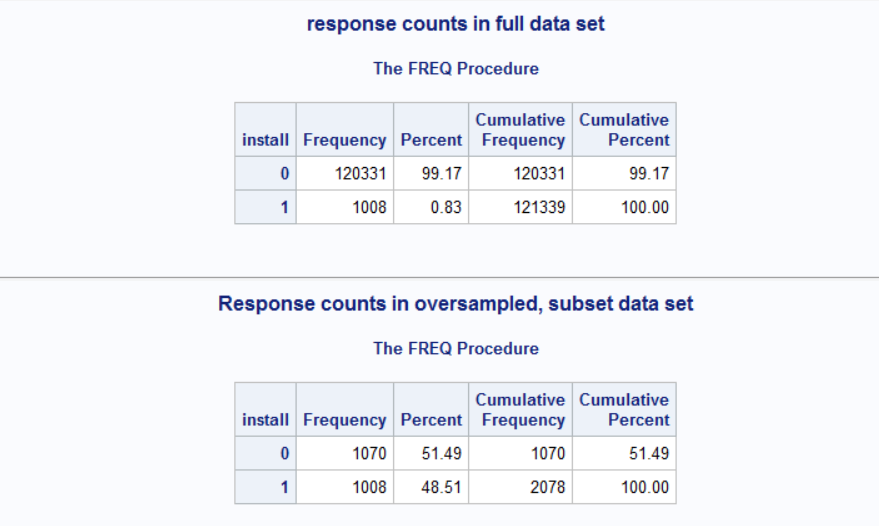
We do not need to compare the number of rare events in this case, because the number of rare events is 1008 in the full sample and 680 in the training data which is high.

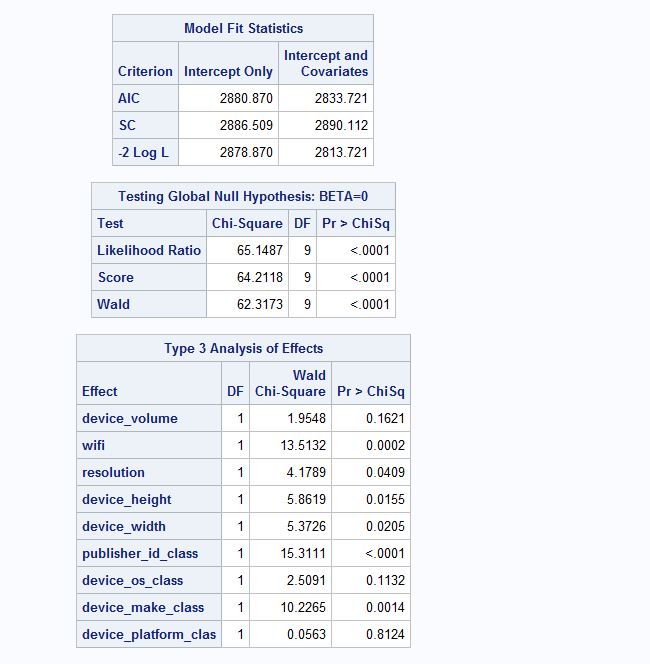
The expected number of rare events should be around 20 for each independent variable which is 10 in this case.

Therefore 200 rare events would have been the optimal count. Since 200<680, the modelling of rare events is not required.

(ii) Estimation of the model considering rare events using oversampling approach to handle rare events and also applying correction mechanism to correct for intercept values.

Using the proc freq statement to get the response counts in the full and over sampled dataset which is the number of rare events.

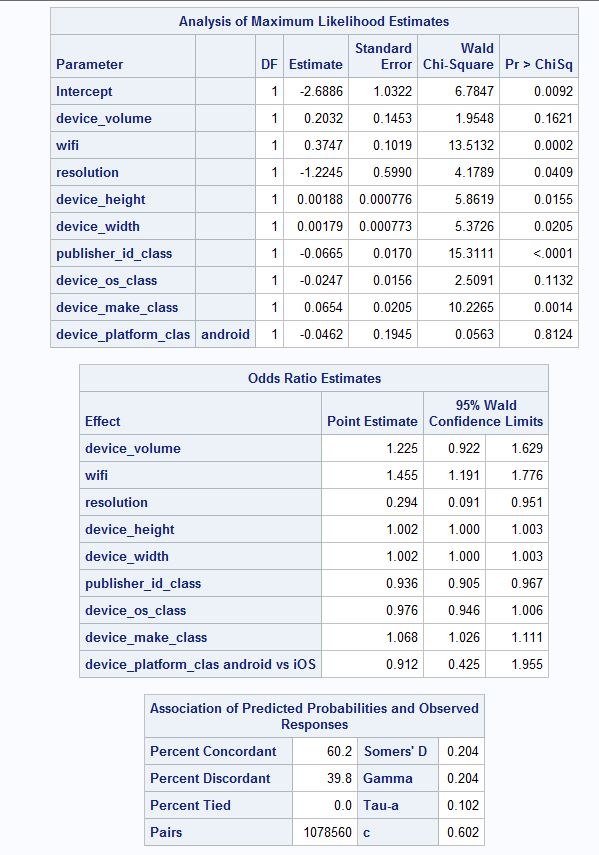




Applying the formula off=log( (r1\*(**1**-p1)) / ((**1**-r1)\*p1) ). to correct for the intercept values.

Unadjusted model:

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,

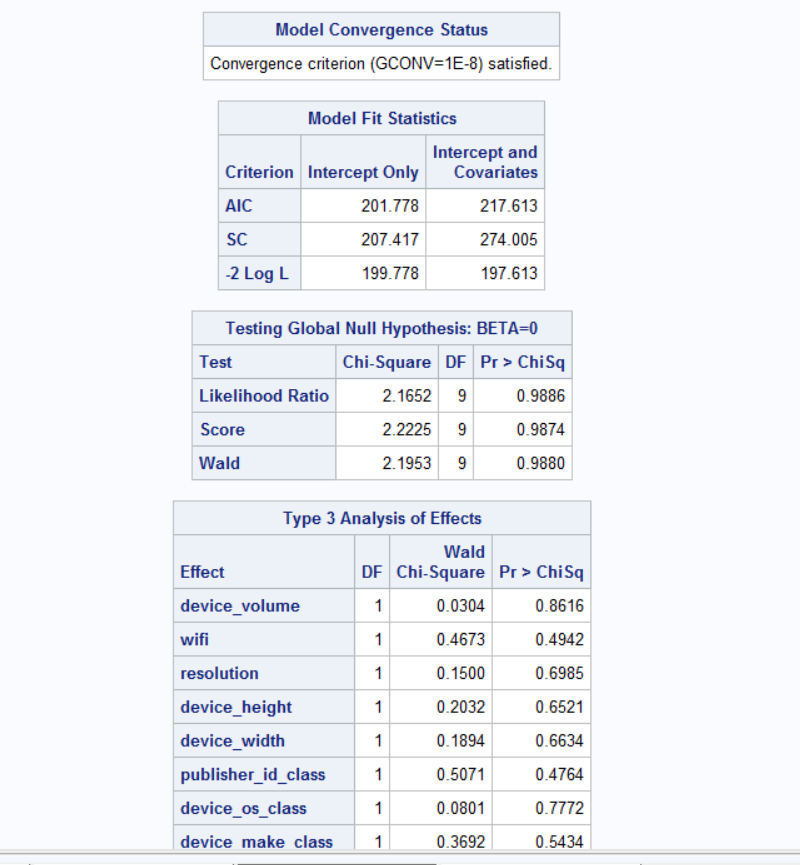
Here we don’t apply the correction, our intercept values deviate from each other.

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.

Weight adjusted model:

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,

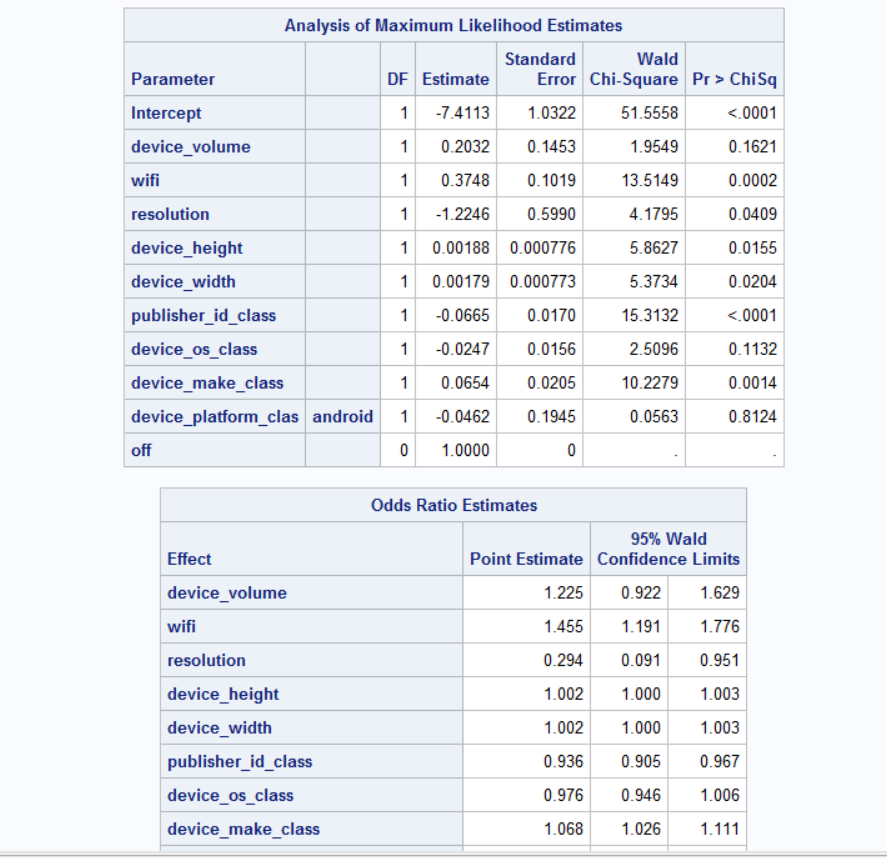
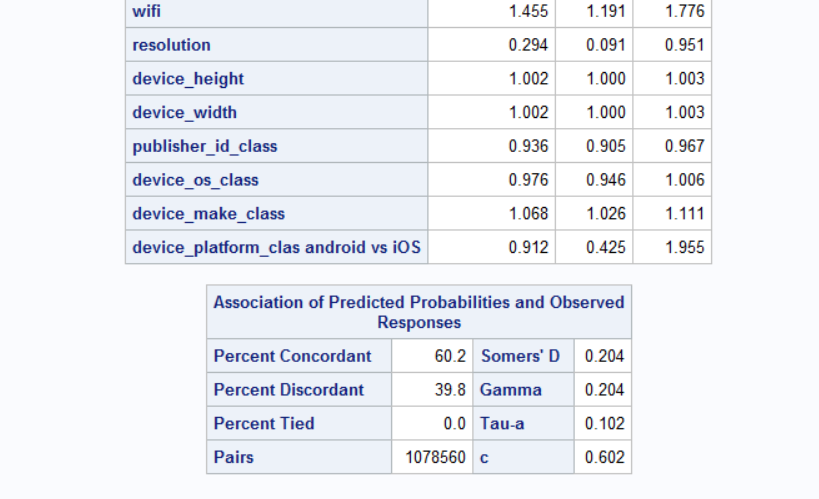
The weight- adjusted model applies the correction which brings the intercept values closer.



Offset adjusted model:

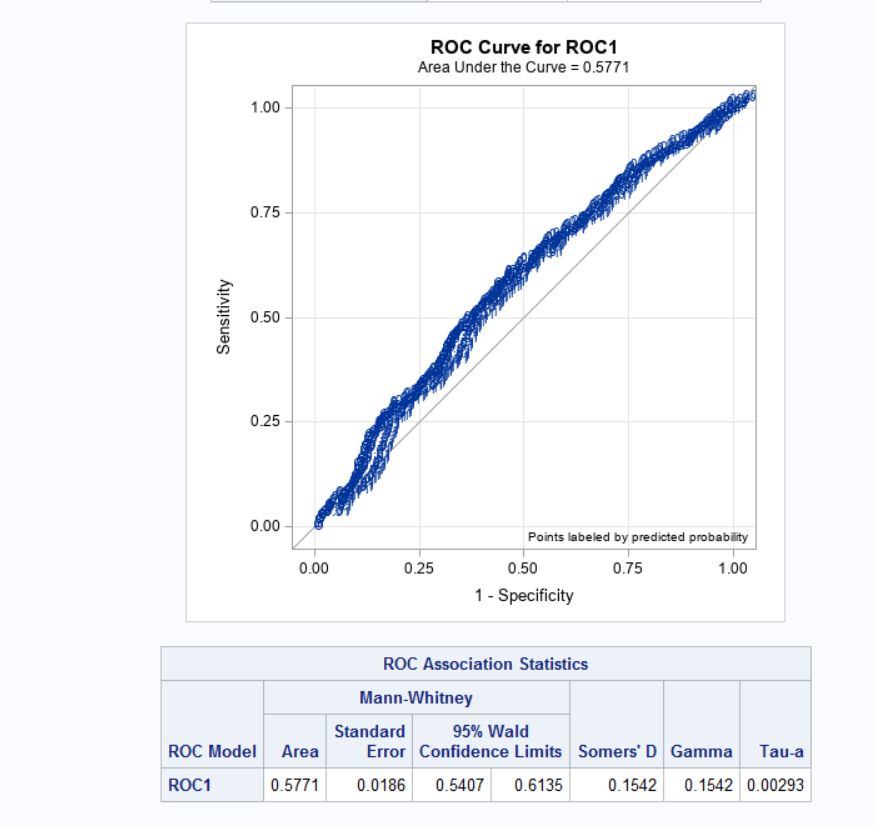
As a secondary approach, we also run the offset adjusted model which yielded the following results,

The next step is to plot the ROC curves for the initial and final linear models and logistic regression models.

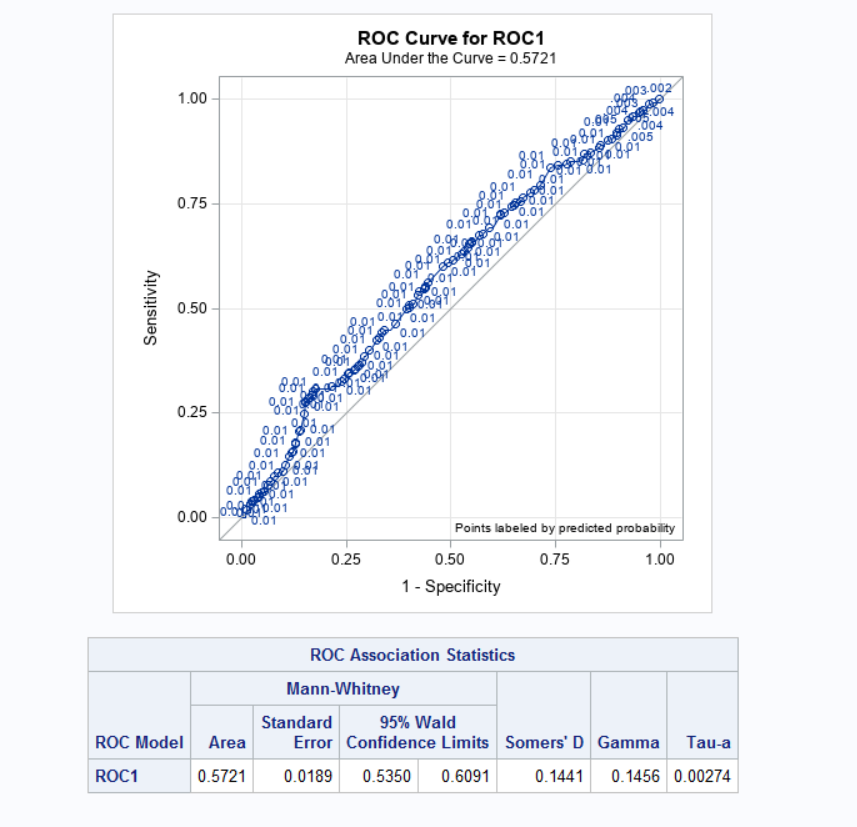
Initial Linear Model:

All the 10 predictors were used to get the below ROC.



Final Linear Model:

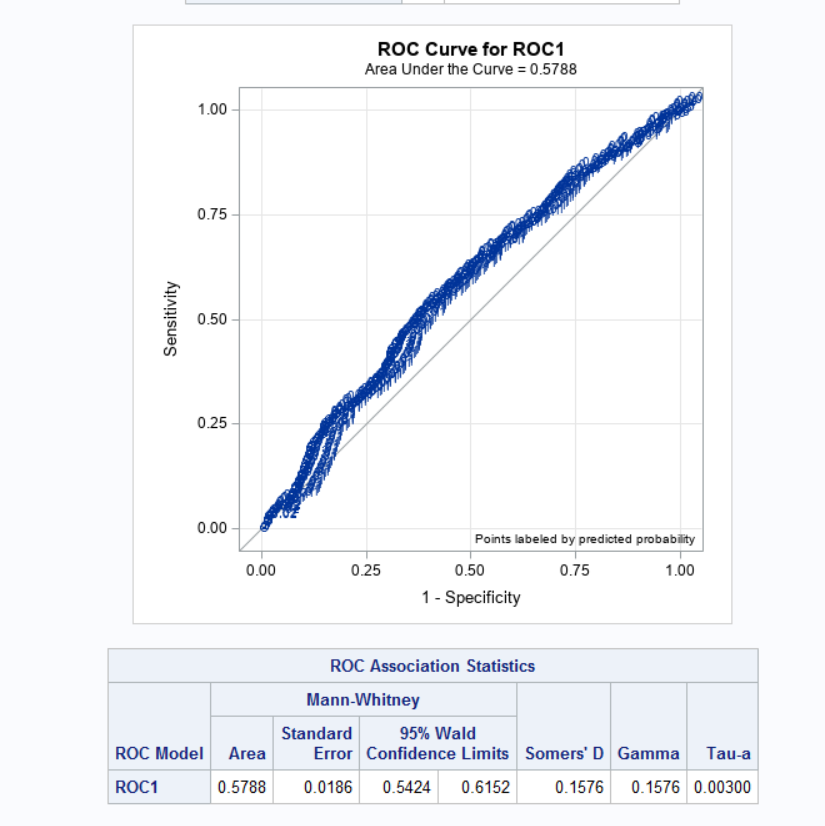
Here getting the ROC with 6 predictors



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.

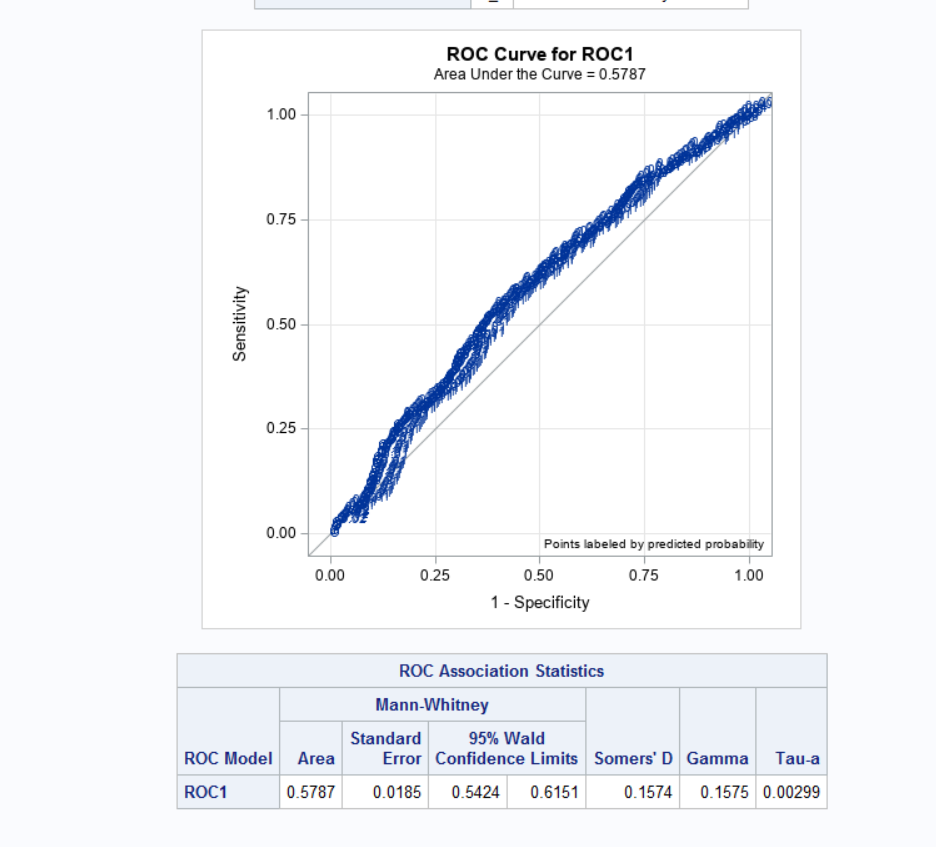
Initial Logistic Model:

Using 10 predictors to get the below ROC Curve



Final Logistic Model:

Using 8 predictors to get the below ROC Curve



Here, the initial and the final logistic models both have ROC curve that looks similar and the area under the curve remains the same.

Conclusion:

From the above results of the ROC Curve; we conclude that the AUC value of Final Logistic Regression = 0.5787 and 95% Conf Interval values ( 0.5424, 0.6151) is higher than the Final Linear Probability Model with AUC = 0.5721 and Conf Interval values ( 0.5350, 0.6091). So we will use Logistic Regression Model for prediction.

Part 2:

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.

First, we need to calculate the total expected cost as follows:

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

The two possible situations where the ad company can incur a loss is as follows:

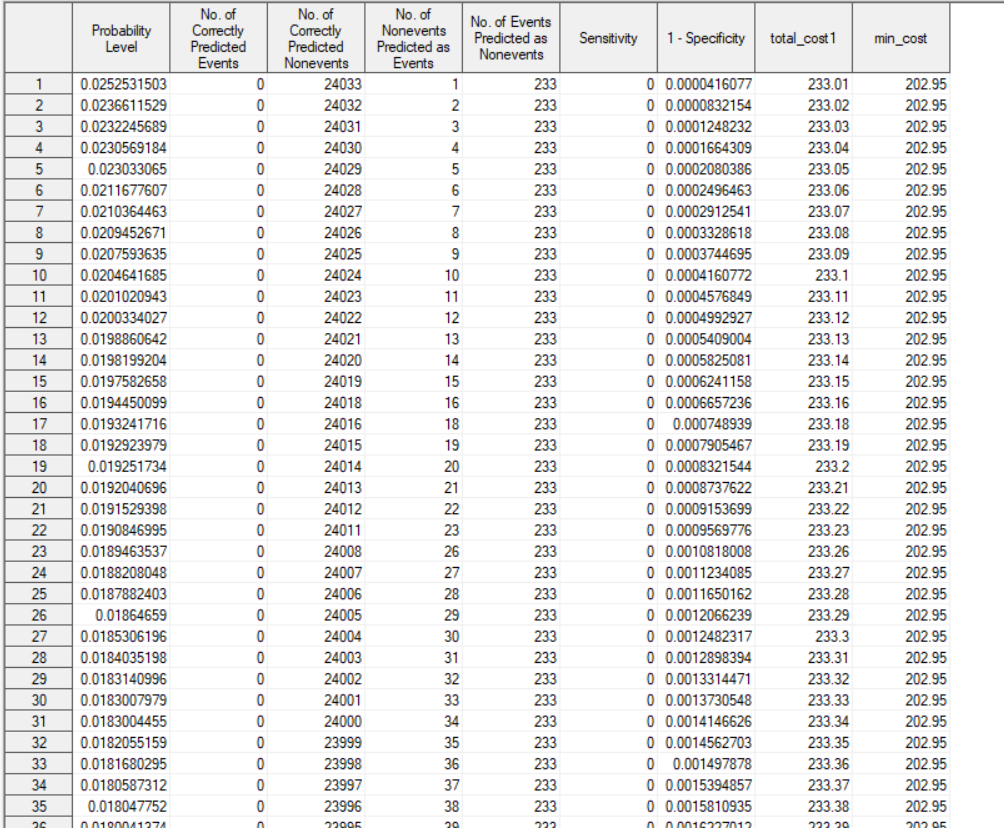
False positive- The platform shows an ad to a consumer but the consumer ends up not installing an app. The loss is estimated to be 1 cent (0.01$)

False negative- The platform fails to show an ad where the consumer actually would have installed the app. The loss here is 1$.

Logistic regression models

The proc logistic statement is used to create a ROC table for both the initial and final models. The total cost column is created using the above formula.

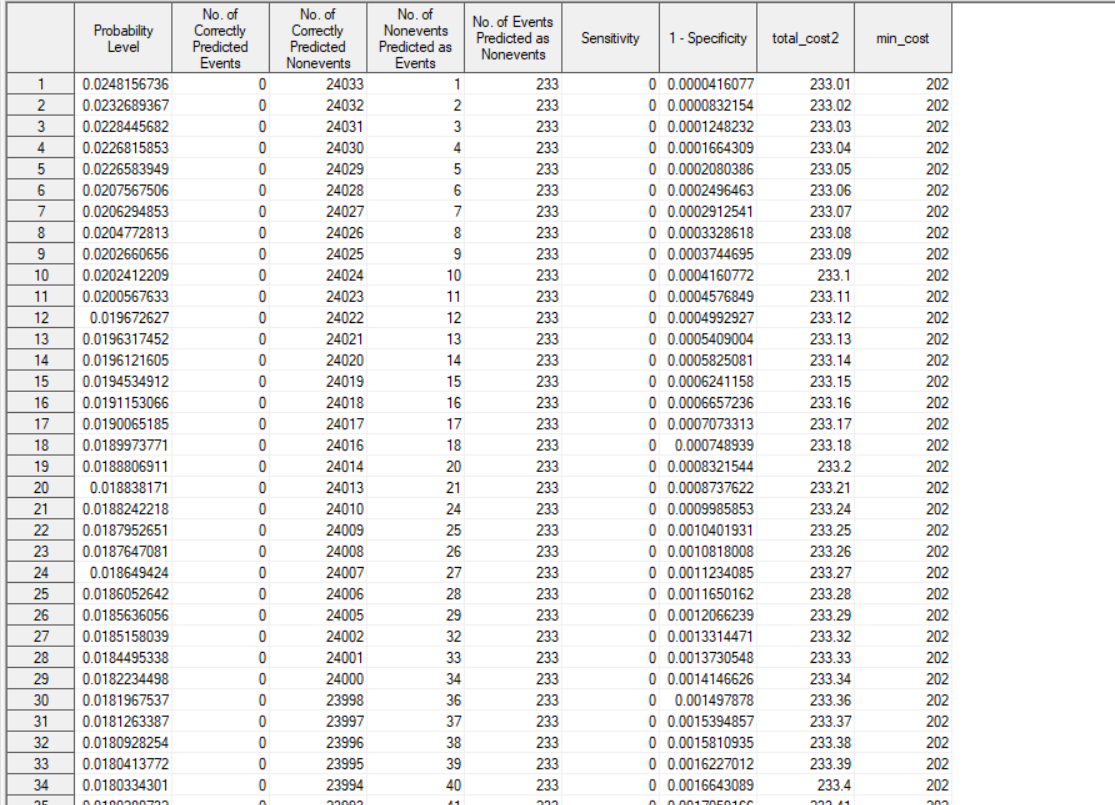
The ROC table for the Initial logistic model is as follows:



Min Cost: $202.95

The probability threshold: 0.0081

The ROC table for the Final logistic model is as follows:



The min cost= $202

The probability threshold= 0.00821

Linear Probability Model:

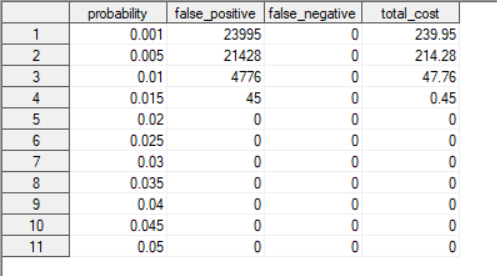
We have to create ROC tables manually for each threshold values.

False positive- if install=0 and predicted=1 then false\_pos=1

False negative- if install=1 and predicted=0 then false\_neg=1

Initial Linear Model:

ROC table for initial linear probability model

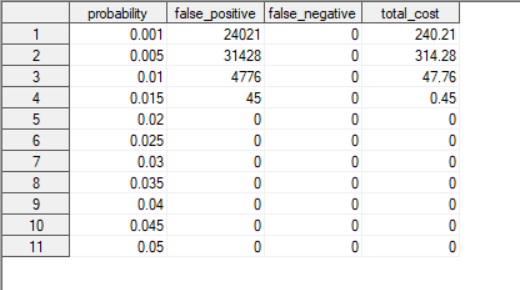


Min cost= 214.28$

The probability threshold= 0.005

Final Linear Probability Model:

We follow the same procedure as we did for the Initial Linear Model:



The min cost= 240.21$

The probability threshold= 0.001

From the above results we can draw the following table:

|  |  |  |
| --- | --- | --- |
| **Model** | **Probability Threshold** | **Minimum total cost** |
| Initial logistic regression model | 0.0081 | $202.95 |
| Initial linear probability model | 0.005 | $214.28 |
| Final logistic regression model | 0.00821 | $202 |
| Final linear probability model | 0.001 | $240.21 |

Therefore, from the results above we can say Logistic Model provide the lowest cost at a probability level 0.0081.