**BUAN 6337 Predictive Analytics using SAS**

**Project 2**

**AADITH NARAYAN RAVISHANKAR / AXR180085**

**INTRODUCTION**

The dataset Involves the domain of mobile advertising. In order to attract customers, the app developer advertises its app on other apps to gain traction. The dataset has the following column description:

|  |  |  |
| --- | --- | --- |
| **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) |

Here, Install is our target variable. Since it is a binary variable, it’s a classification problem. Therefore, we will be using linear probability models and logistic regression to estimate the install variable.

The project consists of two parts:

1. Part One

(i) To develop a linear probability model to estimate the probability of installing the ad based on publisher and consumer characteristics.

(ii) To develop a logistic regression model to estimate the probability of installing the ad based on publisher and consumer characteristics.

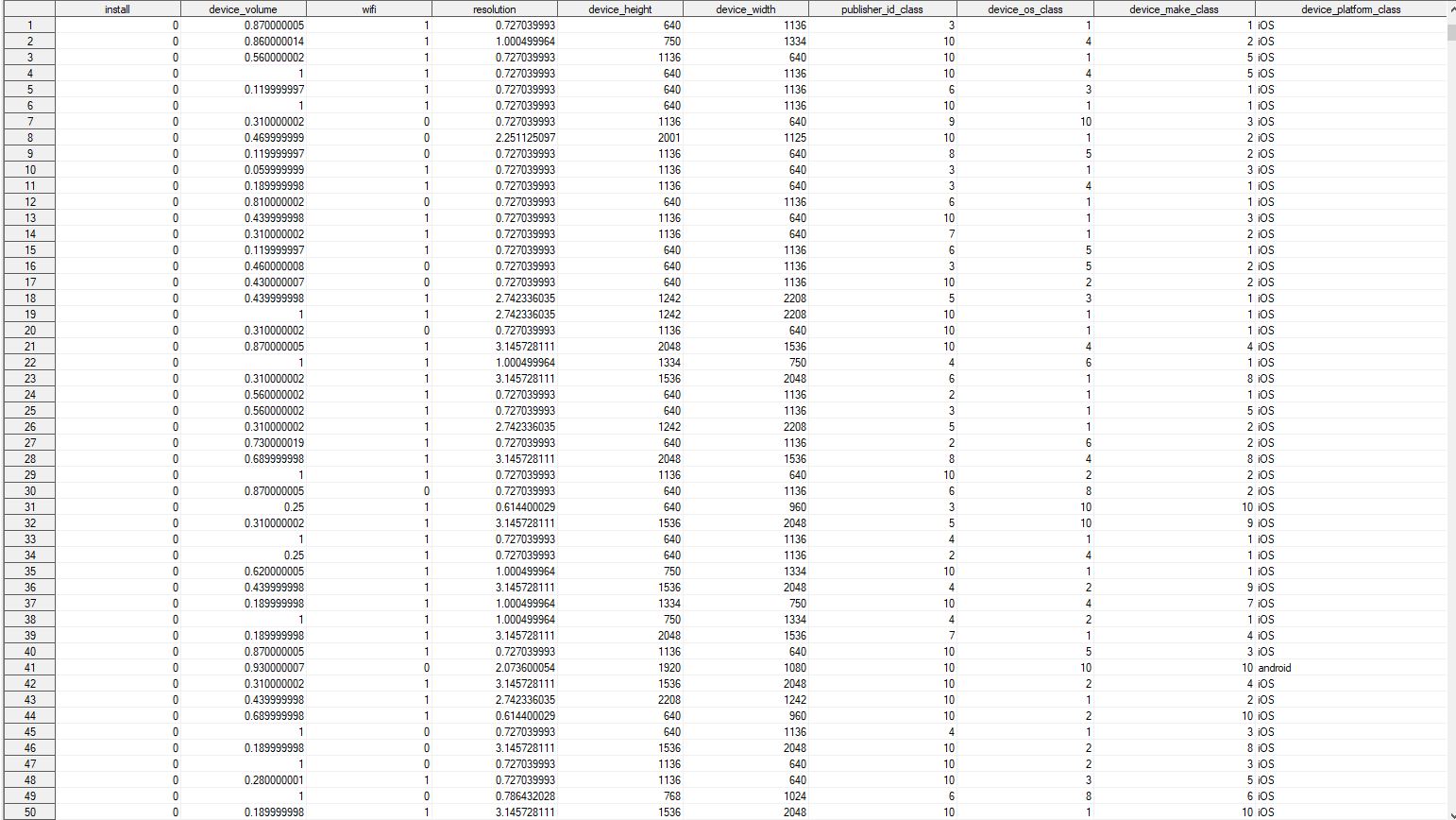
(iii) To produce ROC plots for the above models

2. Part Two

(i) Generating ROC table and plot total cost with respect to threshold values - 0.001, 0.005,0.010,0.015,0.020,0.025,0.030,0.035,0.040,0.045 and 0.050

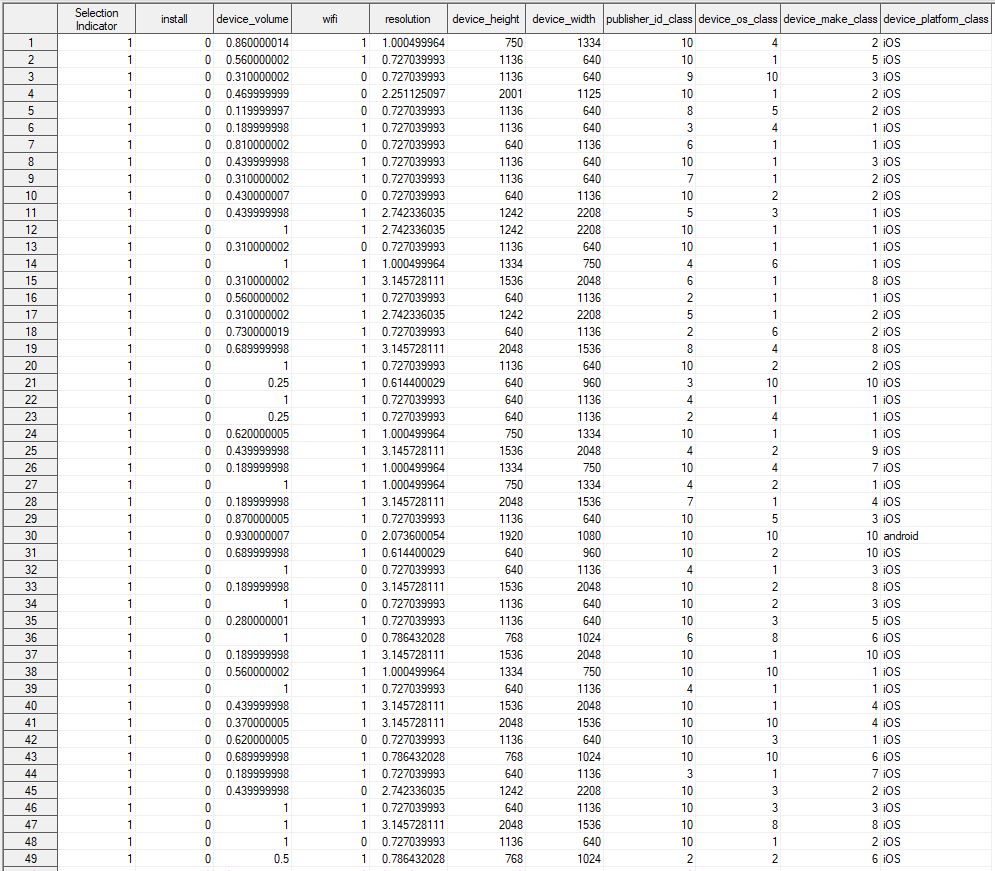
**METHODOLOGY**

* Part One
* Linear probability models

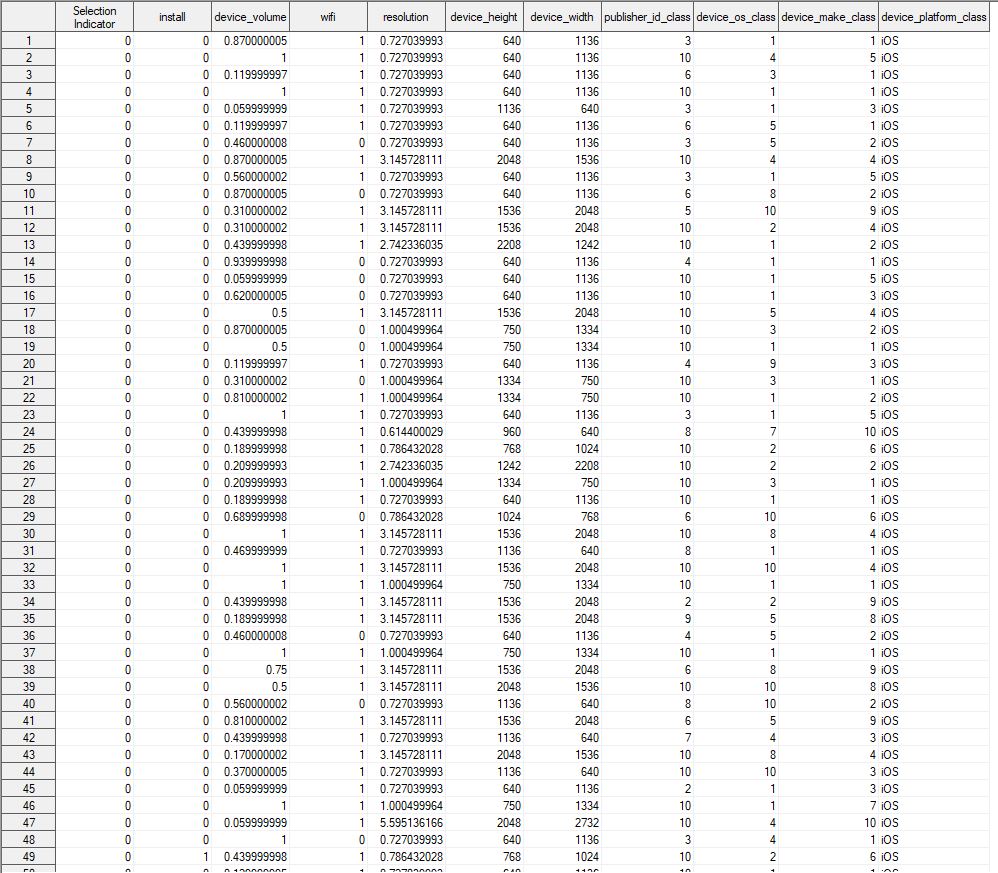
First, we set the working directory using Libname and import the dataset ‘Project2.DATA‘using DATA. The dataset is as follows,

Next, a train test split is done to split the data such that 60% of the data goes to train and 40% of the data goes to test. For that we randomly assign 1 according to a random seed to 60% of the data and filter it out as the training dataset while the rest is defined as the testing dataset.

Training dataset:

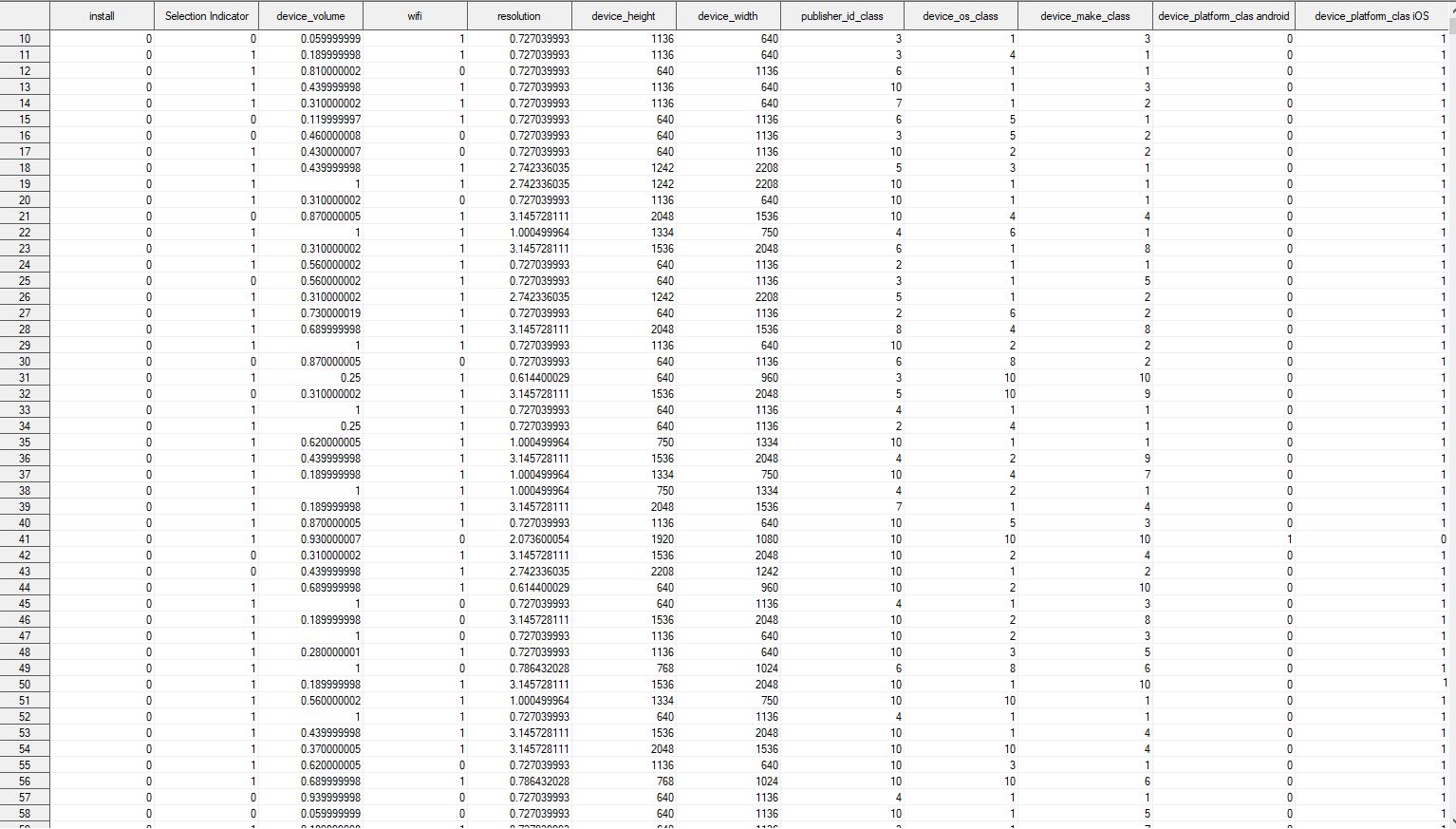


Testing Dataset:

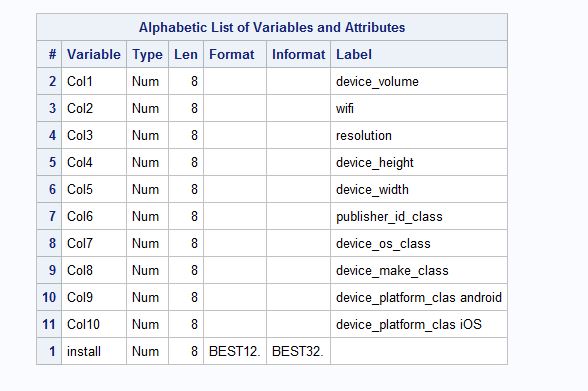


The dataset contains certain categorical variables. As we know, linear models only accept numerical values as input. So, these categorical variables have to be converted to numerical variables. This is done using proc glmmod statement. The conversion is done for the initial dataset, the train and test datasets.

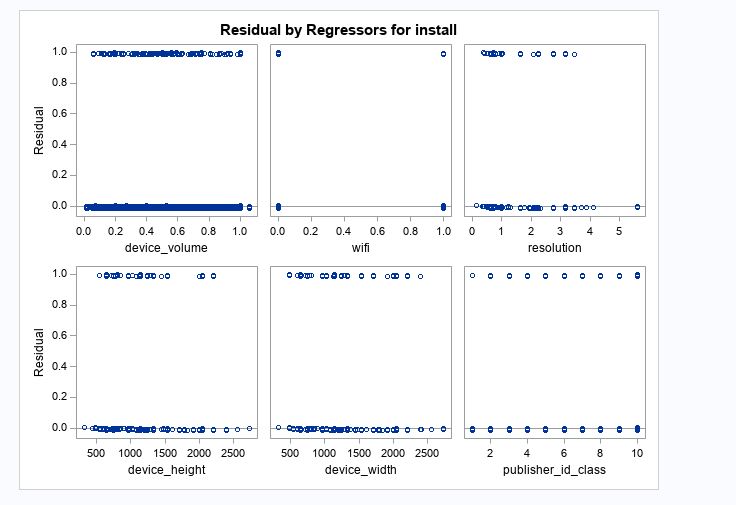
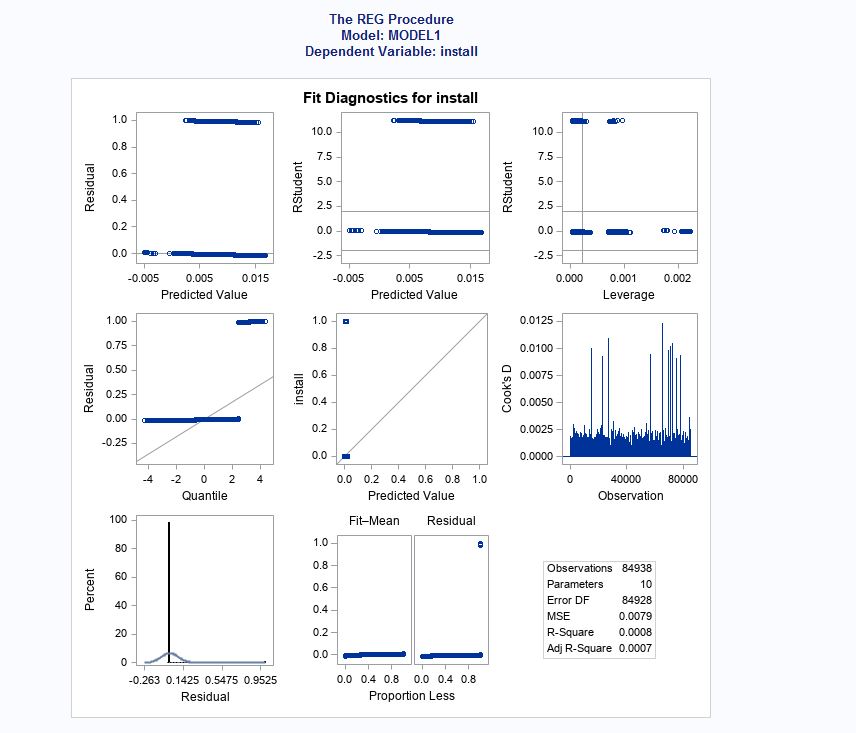
The encoding for the initial dataset. Notice the device\_platform\_class\_ prefixed columns:

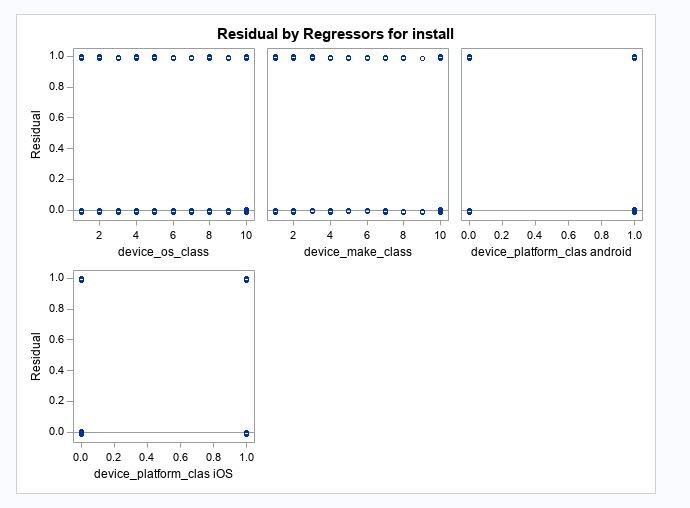


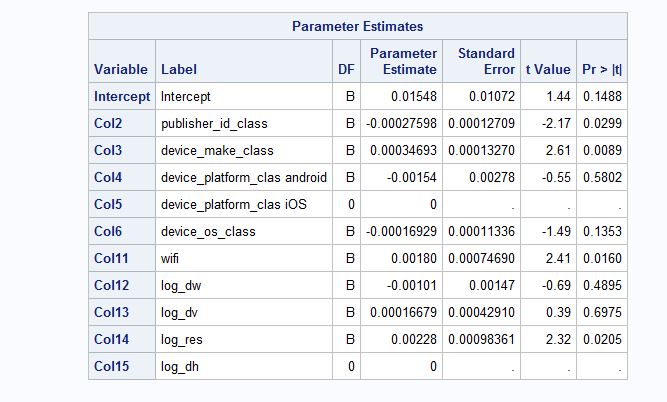
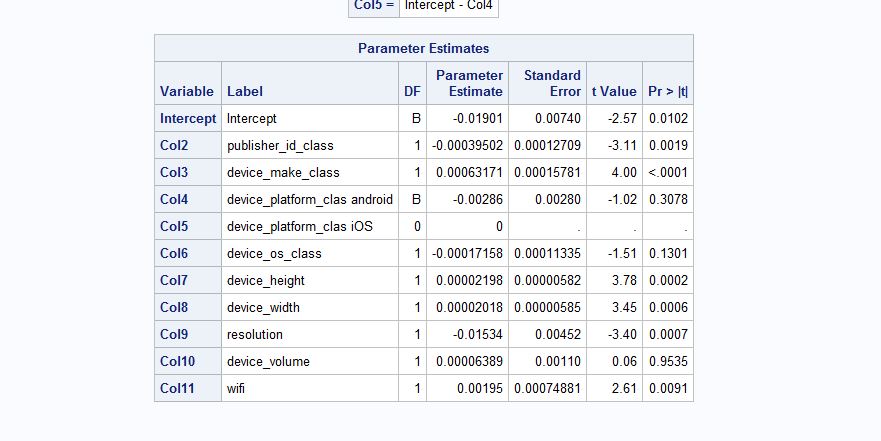
Using the column names every time to define models would be tedious. Therefore, the proc contents statement is used to provide alias names as column numbers to the feature variables.



For instance, device\_volume can be addressed as Col1.

I then use the proc reg statement to run a linear model on all 10 variables as a kitchen sink model with install as the dependent variable. 



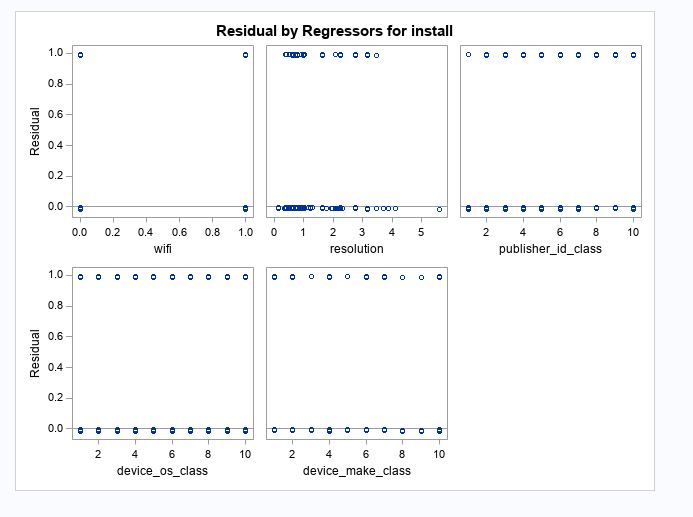
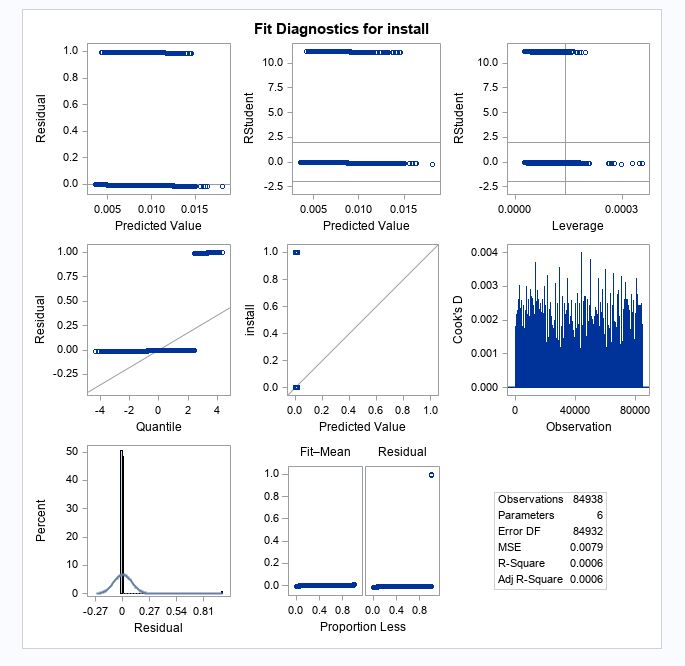
After running the initial model, we see that in the fit diagnostics chart that there are a lot of outliers deviating from the common patterns of the dataset values. Therefore, we make log transformations on the numeric variables which will scale the values to a lower range and compress. This is to make sure that the variance effect of the outliers does not affect the model. A linear model is run using the log transformed variables. 

As we can see the without-log model has more significant variables than the log model when comparing their p-values. Therefore, we go with the without log model.

Now even though we have selected the without log model, we need to choose the best combinations of variables which are suitable to be included in the model. For this I choose to follow a stepwise regression method because it involves both forward and backward selection strategies. 

The stepwise regression gives us the final model which involves the following columns:

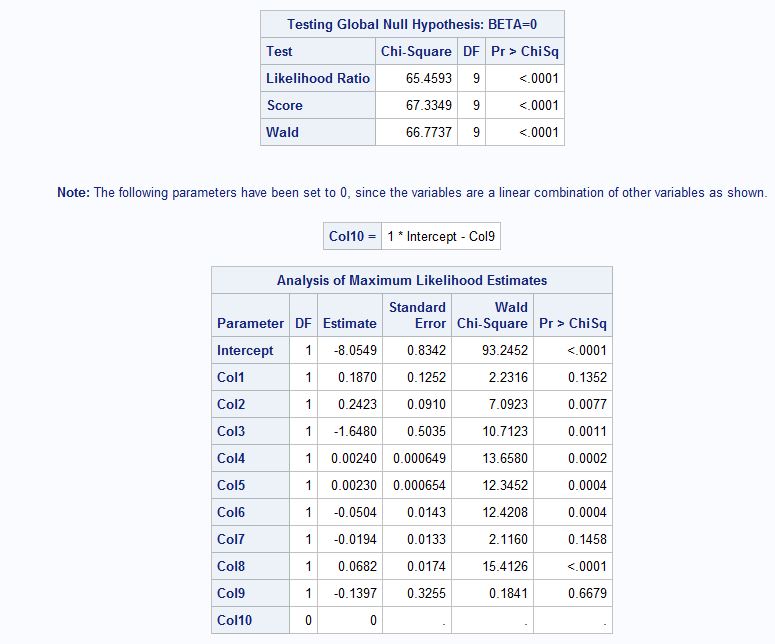
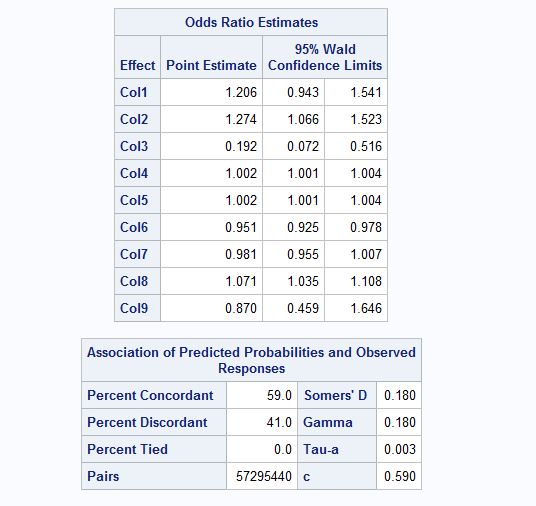
1. Col2- Publisher\_id\_class
2. Col3- Device\_make\_class
3. Col6- device\_os\_class
4. Col7- device\_platform\_clas ios
5. Col8- device\_width

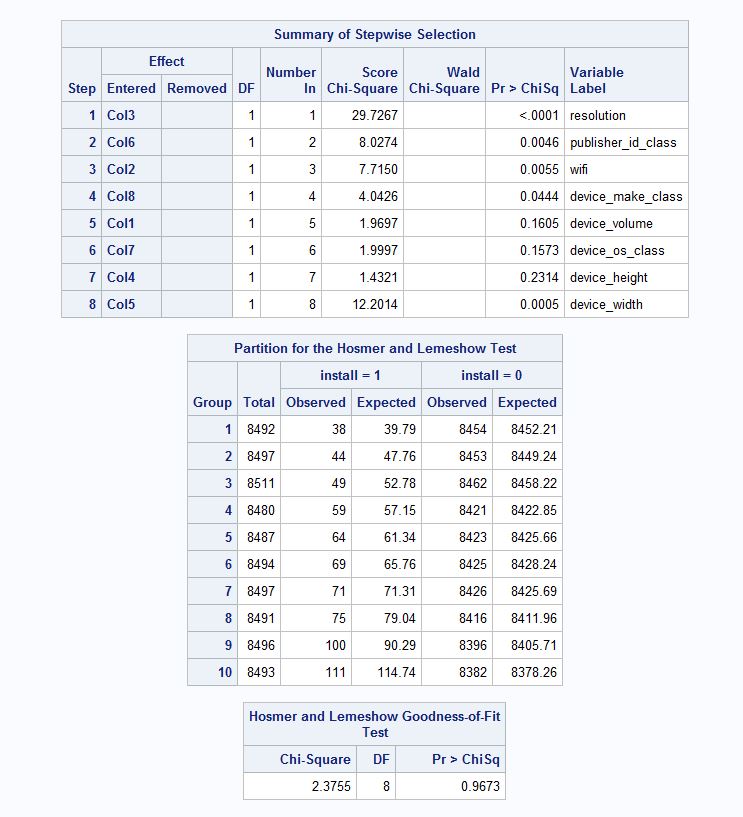
The final model consists of the above columns and a linear model is executed. 

Even if the predicted variables are within probability range, the residuals have to normal which has to be proved using plots. Since the normality assumption is violated, the std error estimates will be invalid and hence the hypothesis testing on predictor variables wouldn’t be valid. A unit change on X does not have the same effect on probability.

* Logistic regression models

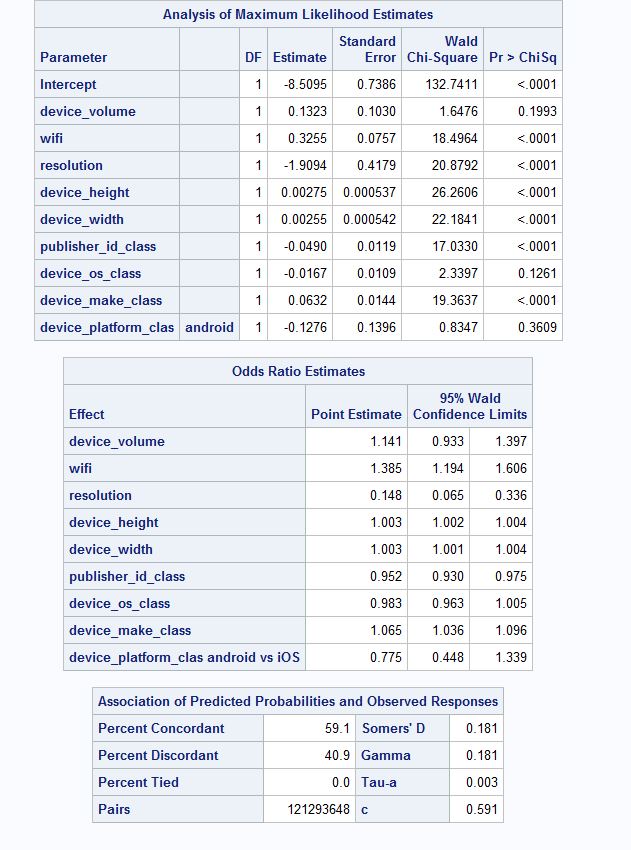
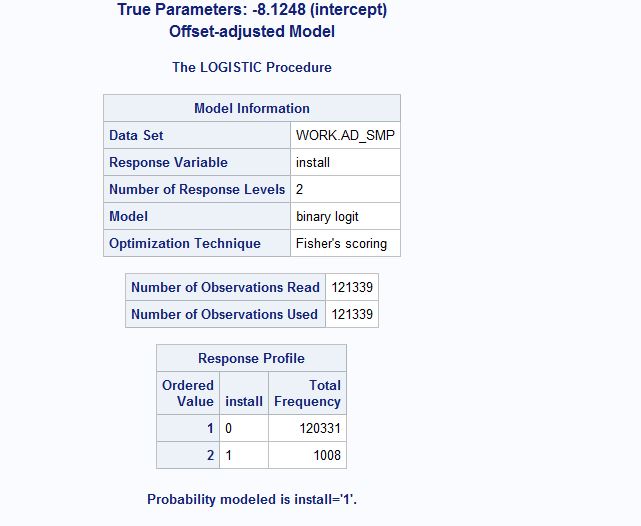
A logistic regression is now run using the same approach. First the initial logistic regression model is run on the 10 columns.



We run a stepwise regression selection procedure to filter out the columns whichare significant to the model. 

The -2 log L values is 7854.786 for the stepwise regression. This is slightly higher than the initial model which had 7854.596.

The columns got from the stepwise regression summary is used to run a logistic regression as the final logit model. There are two versions to do this:

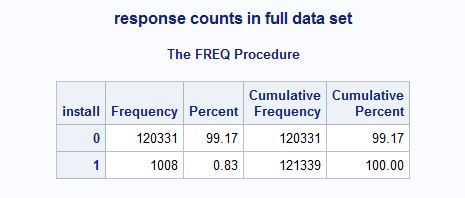
(i)Estimation of the model without considering rare events  

Almost all the variables are significant to the 0.0001 level.

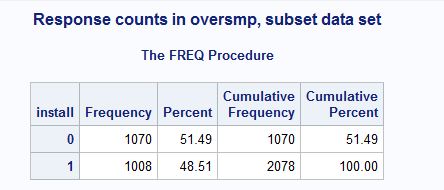
We do not need to compare the number of rare events in this case, because the number of rare events is 1008 in the full sample and 680 in the training data which is high. The expected number of rare events should be around 20 for each independent variable which is 10 in this case. Therefore 200 rare events would have been the optimal count. Since 200<680, the modelling of rare events is not required.

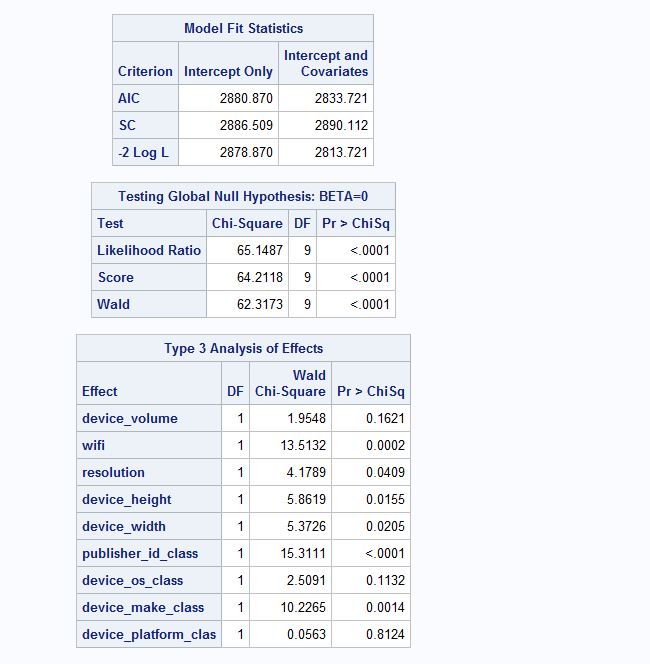
(ii) Estimation of the model considering rare events using oversampling approach and also applying correction mechanism to correct for intercept values.

The proc freq statement to get the response counts in the full dataset which is the number of rare events.



The ratio of the proportions of these rare events to non-rare events is 1008/120331 which is close to 1/119. We then find the response counts in the sampled dataset. The intent is to make the ratio as close to 1 as possible.



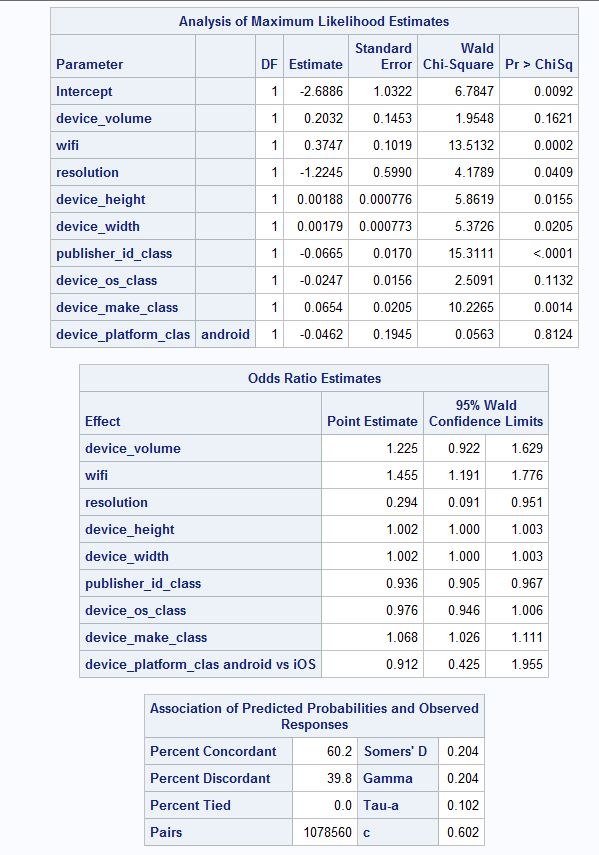
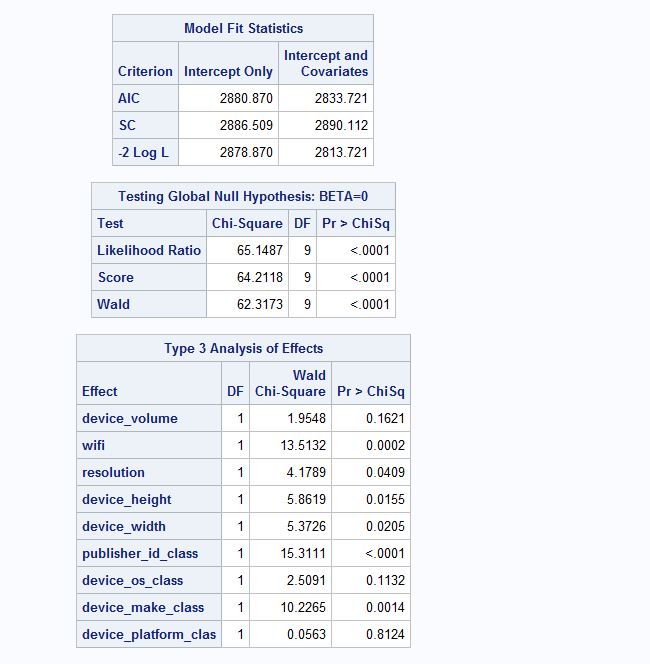


We apply the formula off=log( (r1\*(**1**-p1)) / ((**1**-r1)\*p1) ). to correct for the intercept values.

The unadjusted model- provided we don’t apply the correction, our intercept values deviate from each other.

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

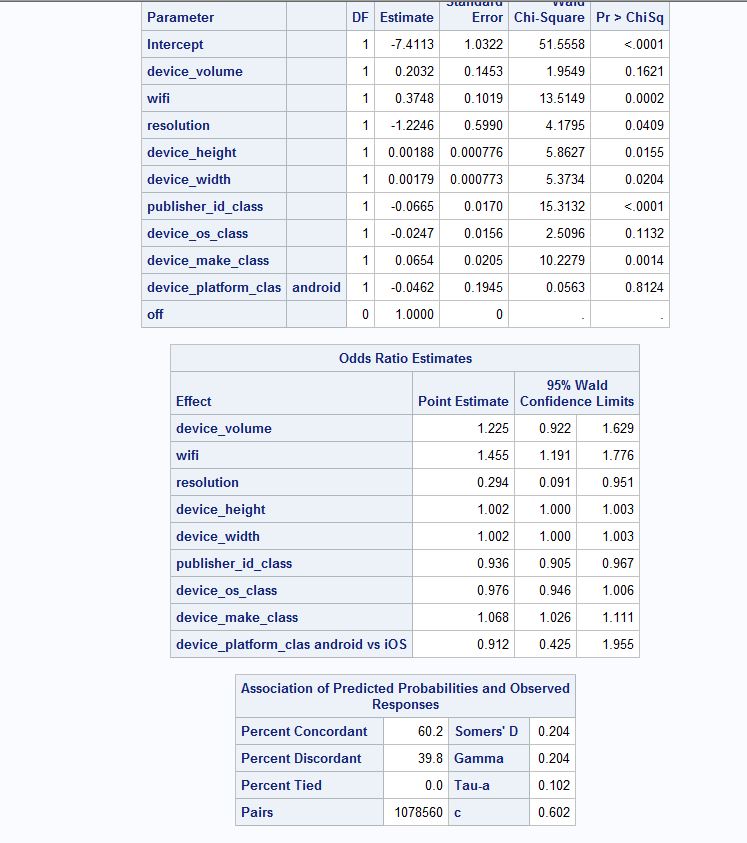
Unadjusted model:



Weight adjusted model:



Offset adjusted model:

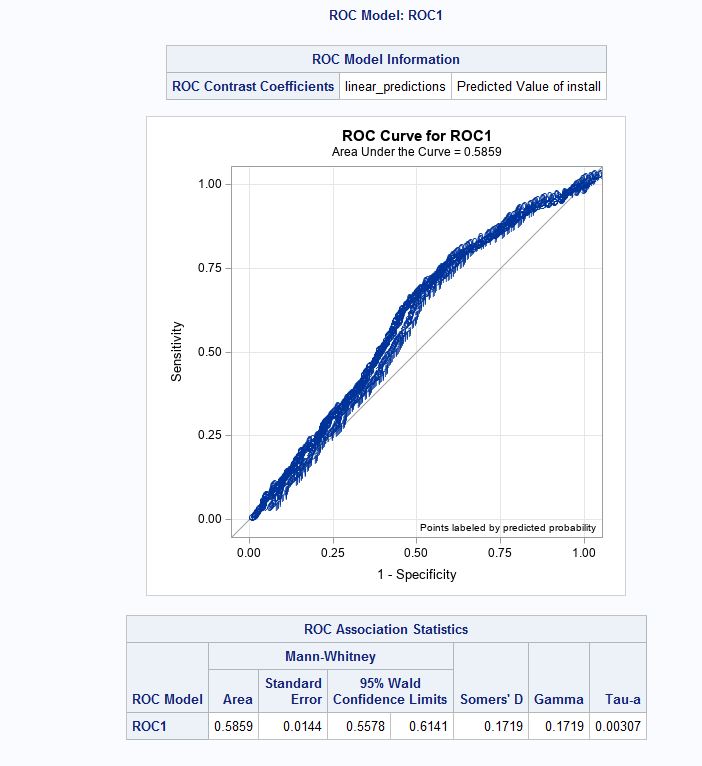


* ROC curves

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

The prediction on the test data is performed for the initial linear model with the 10 columns using the proc reg statement.

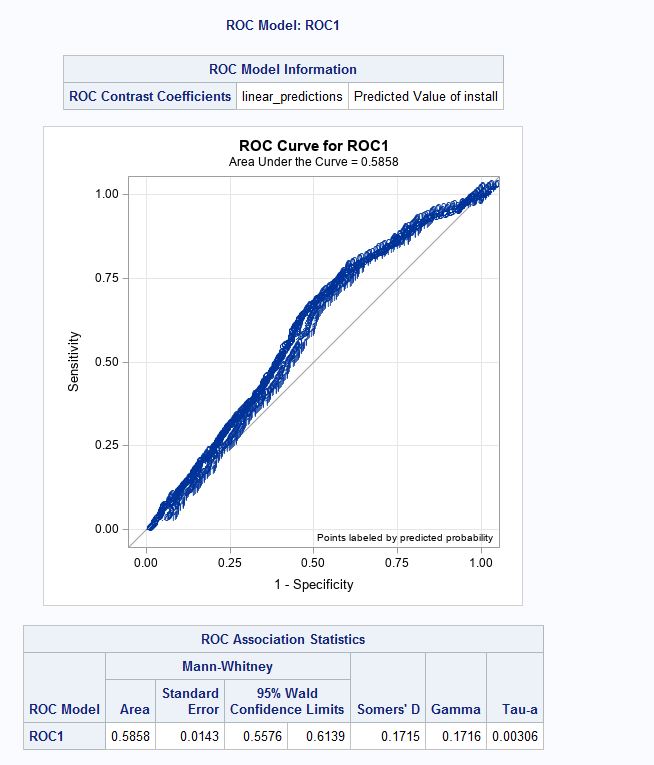
The ROC curve is plotted for the above model using the predicted probabilities as follows:



The area under the curve (AUC)= 0.5859

The prediction on the test data is performed for the final linear model with the selected columns using the proc reg statement.

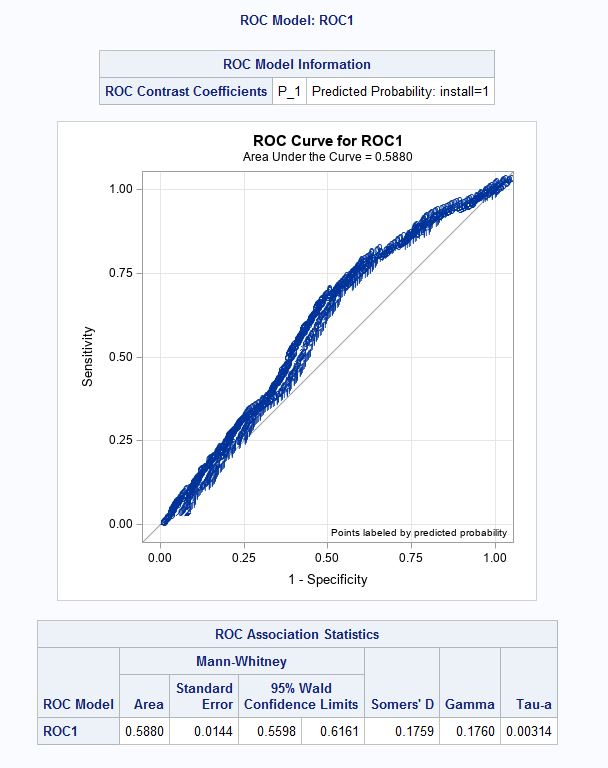
The ROC curve is plotted for the above model using the predicted probabilities as follows:



AUC= 0.5858

The prediction on the test data is performed for the initial logistic regression model with the 10 columns using the proc logistic statement.

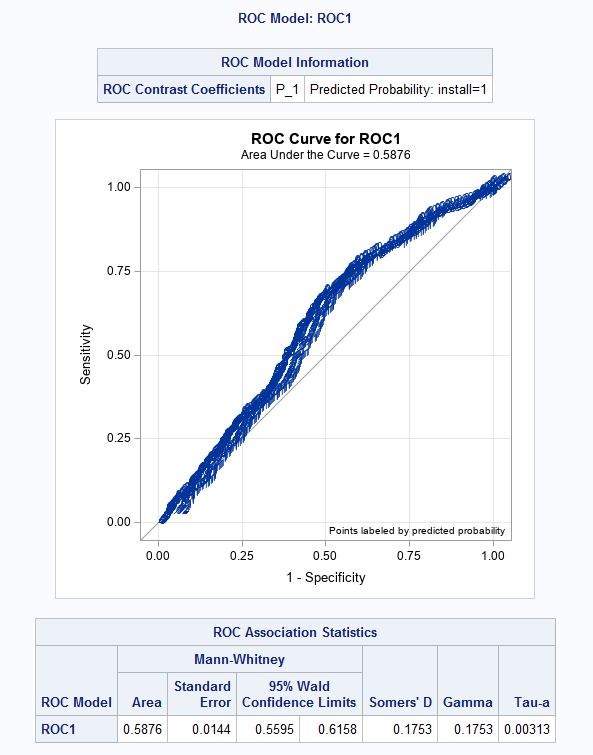
The ROC curve is plotted for the above model using the predicted probabilities as follows:



AUC= 0.5880

The prediction on the test data is performed for the final logistic regression model with the selected columns using the proc reg statement.

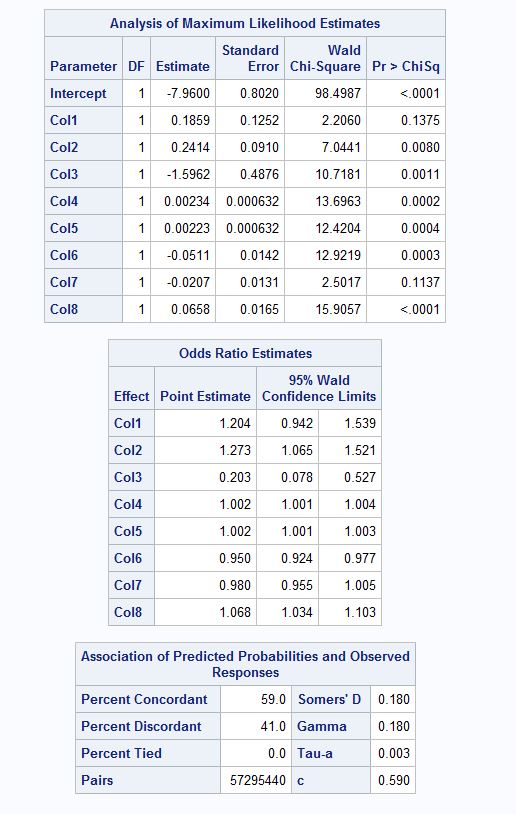
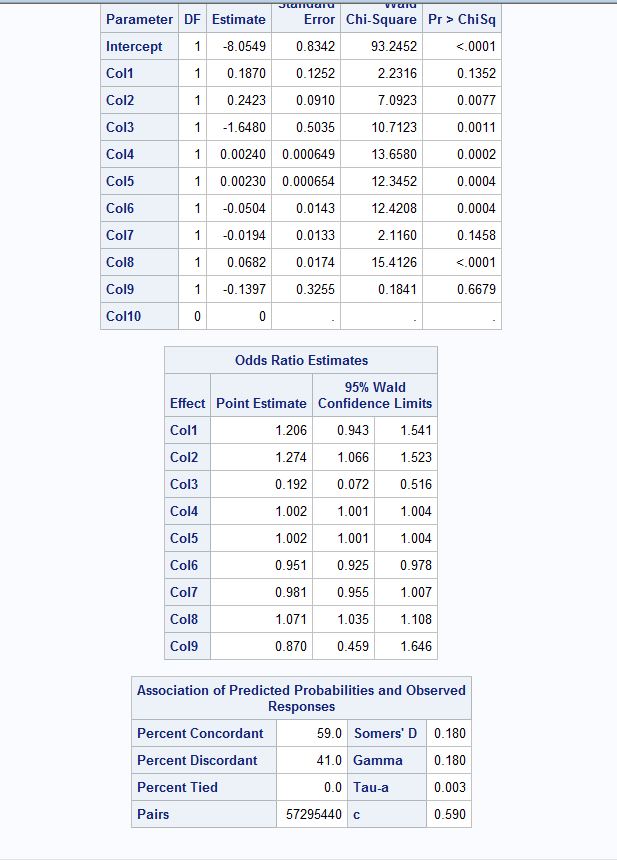
The ROC curve is plotted for the above model using the predicted probabilities as follows:



AUC= 0.5876

**Here, the Area under the curve is highest for the initial logistic regression model and hence we consider that as the final model.**

**Here, the confidence interval range is the highest for the initial logistic regression model which is 0.5598-0.6161.**

The final model is executed with the 10 columns as a logistic regression. 

* Part Two

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:

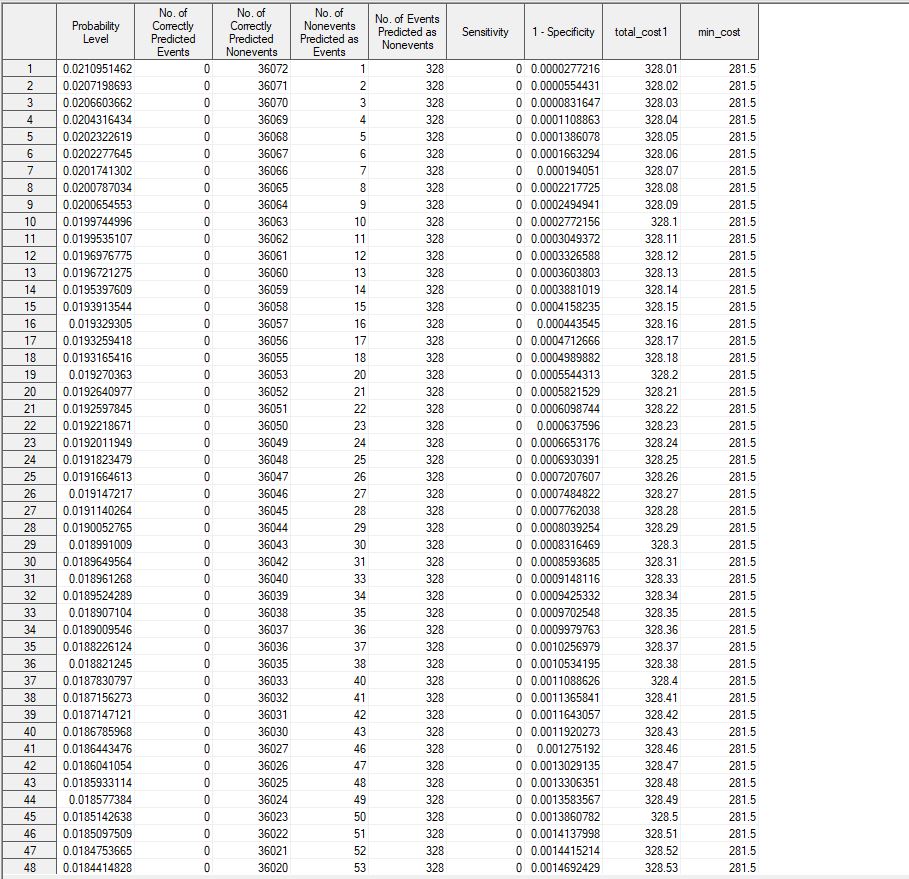
False positive- The platform shows an ad to a consumer but the consumer ends up not installing an app. The loss is estimated to be 1 cent (0.01$)

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.

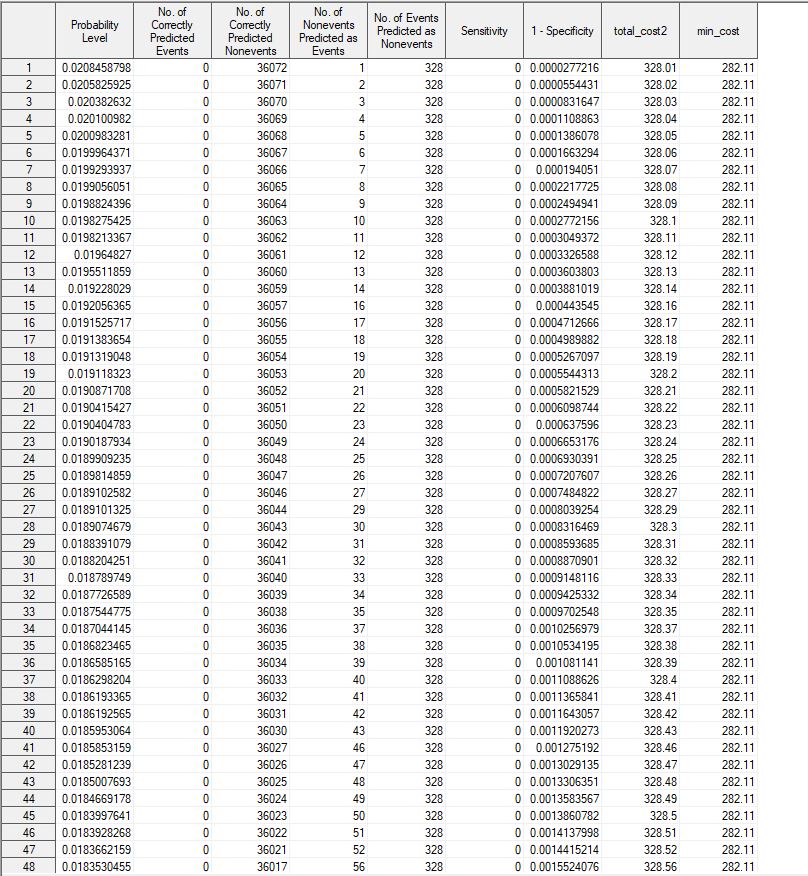
The ROC table for the initial logistic model is as follows:



The min cost= 281.5$

The probability threshold= 0.00753

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



The min cost= 282.11$

The probability threshold= 0.00754

* Linear Probability Model

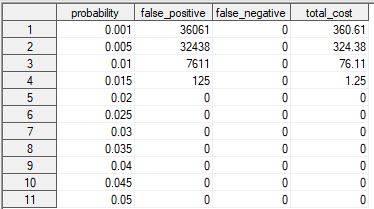
In this case, we have to create the ROC tables manually. Therefore we use the following criteria to define false positives and false negatives:

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

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

The predictions are done only for the test data.

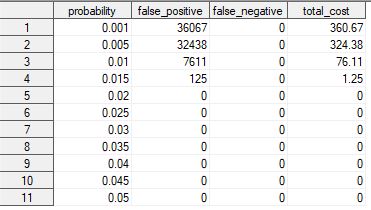
The ROC table for the initial linear model is as follows:



Min cost= 324.38$

The probability threshold= 0.005

The ROC table for the final linear model is as follows:



The min cost= 324.38$

The probability threshold= 0.005

**From the above observations, we can conclude that the initial logistic regression model provides the lowest total cost.**