Lead Scoring Case Study

Problem Statement

An education company named X Education sells online courses to industry professionals. On any given day, many professionals who are interested in the courses land on their website and browse for courses.

The company markets its courses on several websites and search engines like Google. Once these people land on the website, they might browse the courses or fill up a form for the course or watch some videos. When these people fill up a form providing their email address or phone number, they are classified to be a lead. Moreover, the company also gets leads through past referrals. Once these leads are acquired, employees from the sales team start making calls, writing emails, etc. Through this process, some of the leads get converted while most do not. The typical lead conversion rate at X education is around 30%.

X Education has appointed you to help them select the most promising leads, i.e. the leads that are most likely to convert into paying customers. The company requires you to build a model wherein you need to assign a lead score to each of the leads such that the customers with higher lead score have a higher conversion chance and the customers with lower lead score have a lower conversion chance. The CEO, in particular, has given a ballpark of the target lead conversion rate to be around 80%.

Solution

Data reading and understanding

We started with reading and understanding the data.

The first problem at hand we could see was that there were few columns which had a value called 'Select'. On analysing the dataset, we realized this value is for those users who did not select any value from the dropdown of that particular column. And hence 'Select' was as good as null.

Therefore we replaced all select values with nulls in our dataset.

Data cleaning

Next we moved on to Data cleaning which started with handling the null values in each of the columns.

We started by dropping the columns which had more than 45% null values .

After that, we looked for each column having considerable amount of null values one by one.

Our basic strategy to impute nulls for these columns was to first analyse their corresponding values and their frequencies and imputing the nulls with the mode. Also, for certain columns where mode didn't make sense, we imputed the nulls with categories such as 'other' or 'unknown'.

After imputing the null values, we were left with less than 5% null rows which we decided to drop to finally arrive at a dataset with no null values.

Data preparation

After removing the nulls, the next task at hand was to check every categorical column one by one in order to check if we could reduce the number of levels within them. Upon inspecting we could observe that there were various columns having a large number of levels within them, and hence we decided to group them together based on the frequency of each value within those columns.

For example for a column such as Lead Activity, we categorized the value less than 100 into something like ‘other_activity’.

Besides the above, we also noticed that there were several columns (especially the Yes/No) which had little or no variance within them. And hence we decided to drop them from the dataset.

After doing the above operations, we were now ready to prepare our data for modelling.

Creation of dummies

We now moved on creating the dummy variables for all the categorical columns in our dataset. We also mapped the Yes/No columns to 1/0 to convert them to continuous variables as well.

Checking for outliers

We then checked for outliers in continuous variables 'TotalVisits','Total Time Spent on Website','Page Views Per Visit' and removed the significant outliers from the dataset.

Feature Standardization

Finally we standardized our dataset so as to get all the variables upon common scale for model building

We also checked the lead conversion rate which was approximately 38%

Now we were all set for model building.

Model building

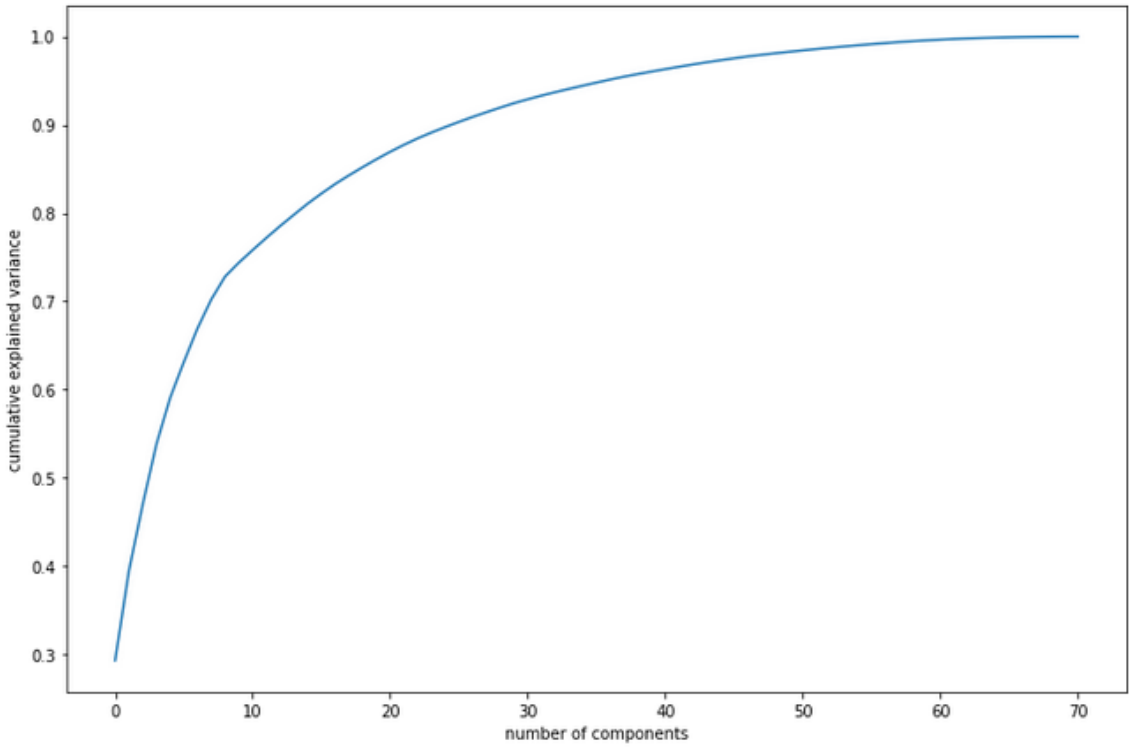
After splitting the data into train-test sets, we decided to build two logistic regression models – one using Principal Component Analysis and the other using RFE.

The results of the model with PCA were slightly better than without PCA and hence here we will only talk about the model that we will finally use. (model with PCA)

Model with PCA

We used the PCA module of the sklearn library to apply PCA on our dataset.

We plotted the screeplot in order to arrive at an optimum number of principal components (PCs).



From above, we can see that 40 PCs can explain more than 90% of the variance in the dataset and hence we chose 40 as the number of PCs for further analysis.

After this, we performed the basis transformation and got our data onto the PCs to finally apply the Logistic regression on the same.

We utilized the LogisticRegression method of the sklearn library to build our model and finally get the predicted values

We created a new dataframe to get predicted probabilities and then the new predicted column by choosing a default cut off of 0.5 for lead conversion.

Then we checked all the evaluation metrics on the training set and following were our observation

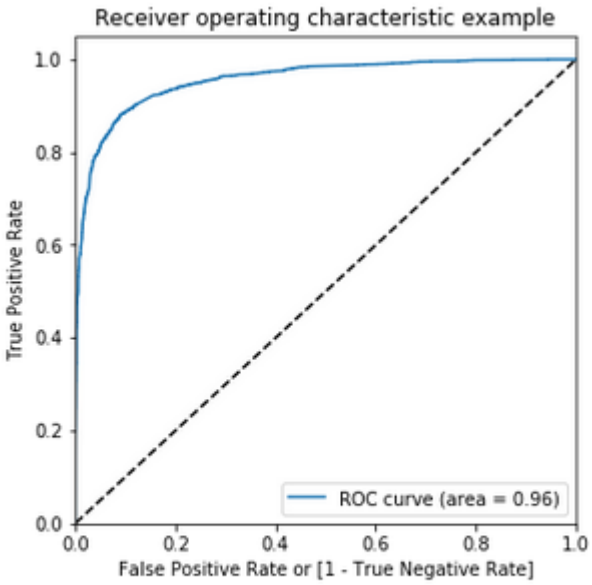
Accuracy – 90%

Sensitivity – 84%

Specificity – 93%

Positive predictive value – 89%

Then we went on to plot the ROC curve as below:



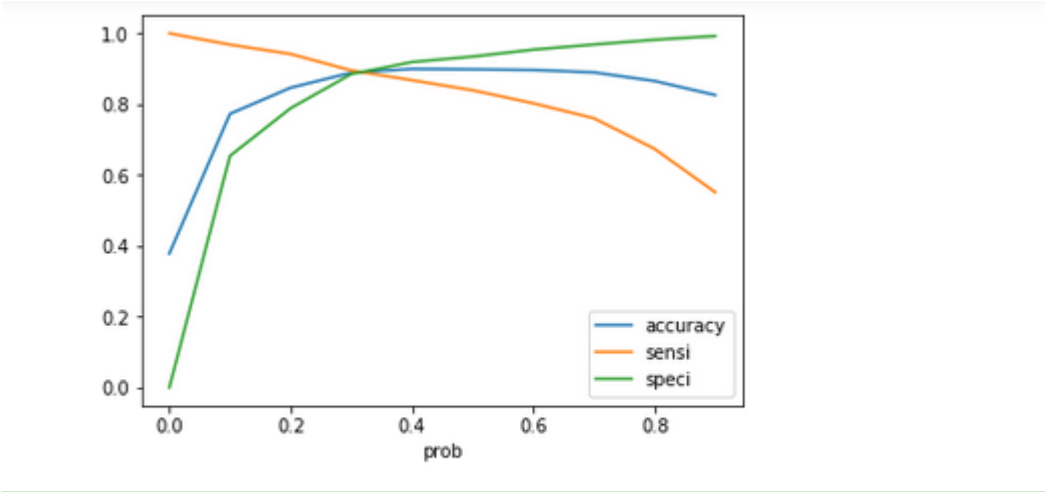
ROC curve looks very good wity auc = 0.96

Next task was to find the optimal cutoff as above we had only chosen a default cutoff of 0.5

Finding optimal cutoff

For this we created separate columns for each probability cut off and also calculate accuracy sensitivity and specificity for various probability cutoffs.

Finally we plotted accuracy sensitivity and specificity for various probabilities to arrive at an optimal cutoff.



From the curve above, 0.3 is the optimum point to take it as a cutoff probability.

After applying the above cutoff to our training set, we observer the evaluation metrics again and following were the figures:

Accuracy – 89%

Sensitivity – 90% (well above the CEO's ballpark figure of 80%)

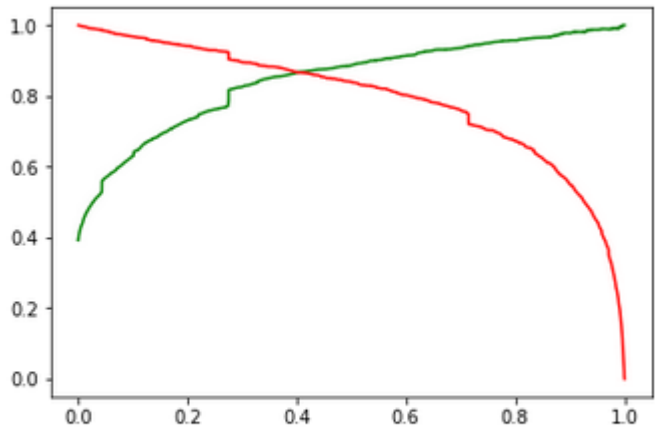
Specificity – 88%

We then finally obtained the lead score on training set by multiplying the predicted probabilities by 100.

Besides above, we also check the Precision-Recall curve to arrive at optimal cut off.

Precision – 89%

Recall – 84%



From above, we can see probability cut off should be approx 0.4

Making predictions on test set

Now we used the above trained model to make predictions on our test set using probability cut off of 0.3 and arrived at final evaluation metrics on test set as below:

Accuracy – 89%

Sensitivity – 88% (which is perfect looking at 80% target that we were initially given)

Specificity – 89%

Finally we got the lead score for each row on test set as well by multiplying the predicted probabilities by 100.

Beside above, we also built another model using RFE but we saw that the sensitivity achieved by this model was 86% and therefore higher than what is expected by the CEO(80%) but is still less than what we got with the model using PCA(88%) as above. Hence we chose the Logistic regression model with PCA for finally making new predictions on new test sets for X Education as that better serves the business problem