LEAD SCORING CASE STUDY

SUBMITTED BY:

1. **MANISH BHARATI**
2. **ANUSHKUMAR K**

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%.

STEPS AFTER READING THE DATA

Data Inspection:

- Data is read as a .csv file as the first step with proper encoding
- Data is inspected for its shape, describe and columns

Data Cleaning:

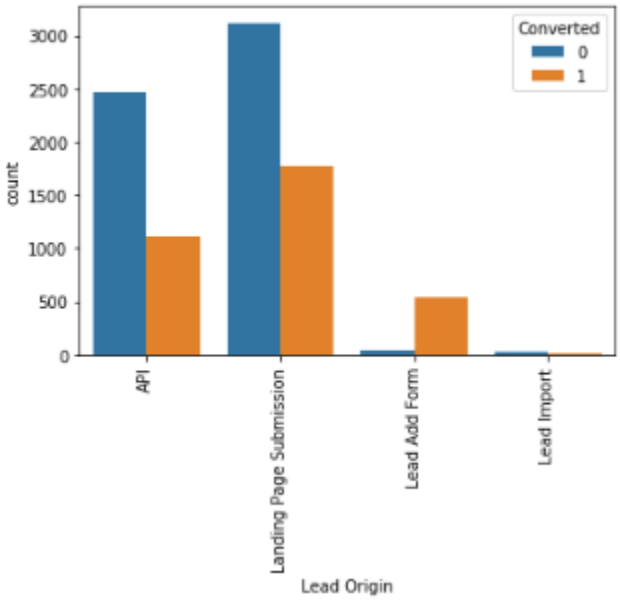
- Data is checked for any missing values, null values
- If there are null values in the data set, then they are being imputed with the median or mode based on the type of variable.
- In our Leads Dataset we have another value as "**Select**" which is also imputed with a value.
- This value corresponds that the user would not selected the field or the field value is not provided in the front end.
- Few columns dropped from Dataset due to high variations:

    1. Asymmetrique Activity Index
    2. Asymmetrique Activity Score
    3. Asymmetrique Profile Index
    4. Asymmetrique Profile Score

Univariate Analysis with converted target variable:

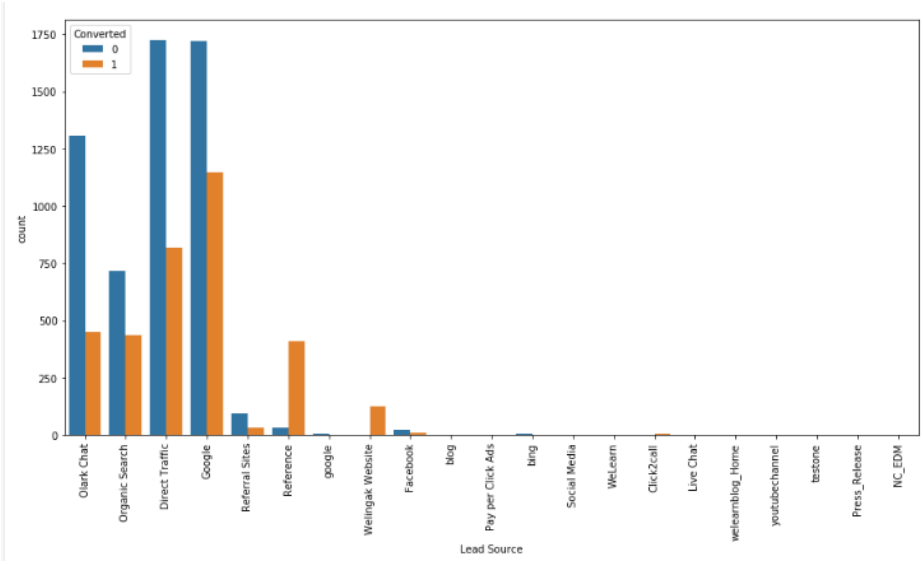
- The current Lead conversion rate is **38%** as per the dataset.

Lead Origin vs Converted



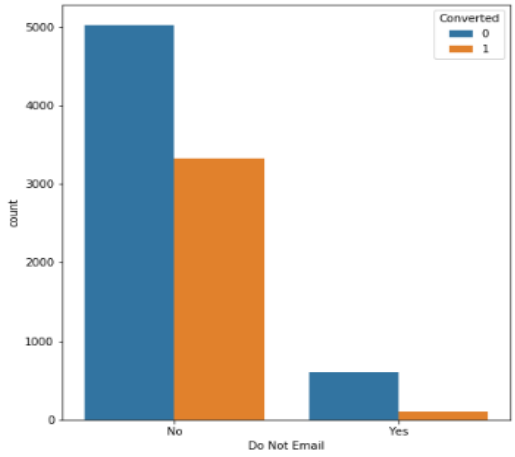
- When Lead Origin is compared with converted variable we observe that the conversion rate is around **30-35%** via API and Landing page.
- But the count of the Leads are very less through **Lead import** and **Lead Add Form** which needs to be focused now.

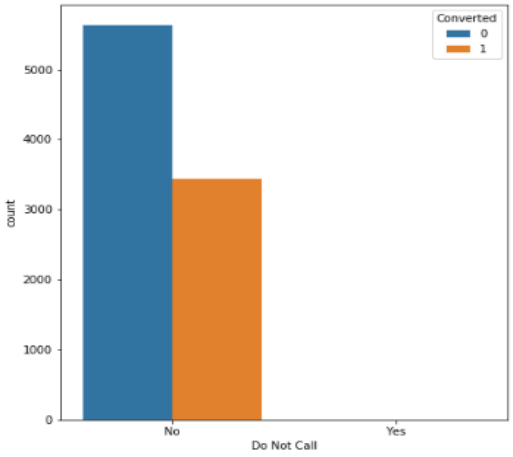
Lead Source vs Converted



- The count of the leads is maximum from **Google Ads, Direct Traffic, Organic search, Olark chat**

Do Not Email vs Converted:



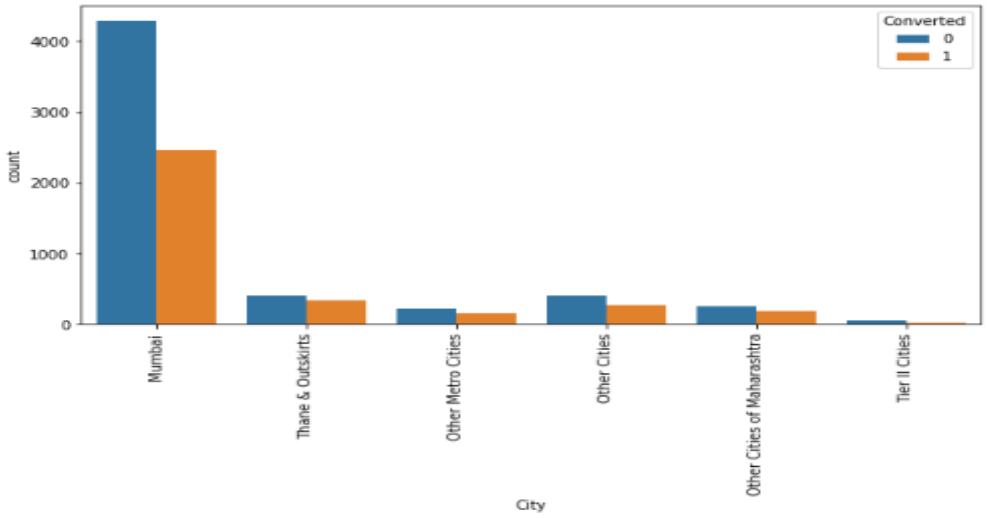


- Those customers who opt for the option of Do Not Email option as "**No**" get converted to Leads

Do Not Call vs Converted:

- Those customers who opt for the option of Do Not Call option as "**No**" get converted to Leads

City vs Converted:

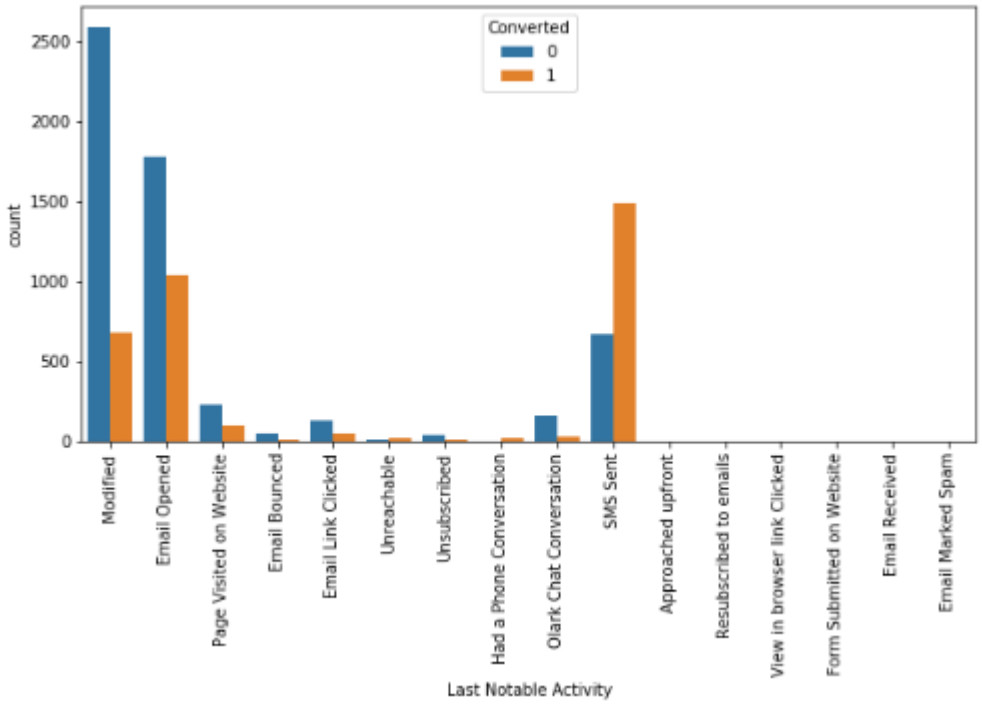


- Most Leads who are converted are from the city of **Mumbai** than any other cities.

Last Activities vs Converted:

- Those customers whose last activity is **reverting back to the email** get mostly converted to Leads.

Last Notable Activity vs Converted:



- Customers whose last notable activity **as Email opened and SMS Sent** are mostly converted to Leads.

CONCLUSION FROM EDA:

- Based on the univariate analysis we have seen that many columns are not adding any information to the model, hence we can drop them for further analysis
- Few columns such as, **What matters most to you in choosing course,Search,Magazine,NewspaperArticle,Through**

Recommendations etc. are the columns which does not give significant conclusions through univariate analysis.

DATA PREPERATION:

- Initially we convert few variables which have only the binary outcomes such as **Yes/No to 1/0** which will be useful while analyzing the data
- We then create the dummy variables for the **categorical columns with multiple levels** of values

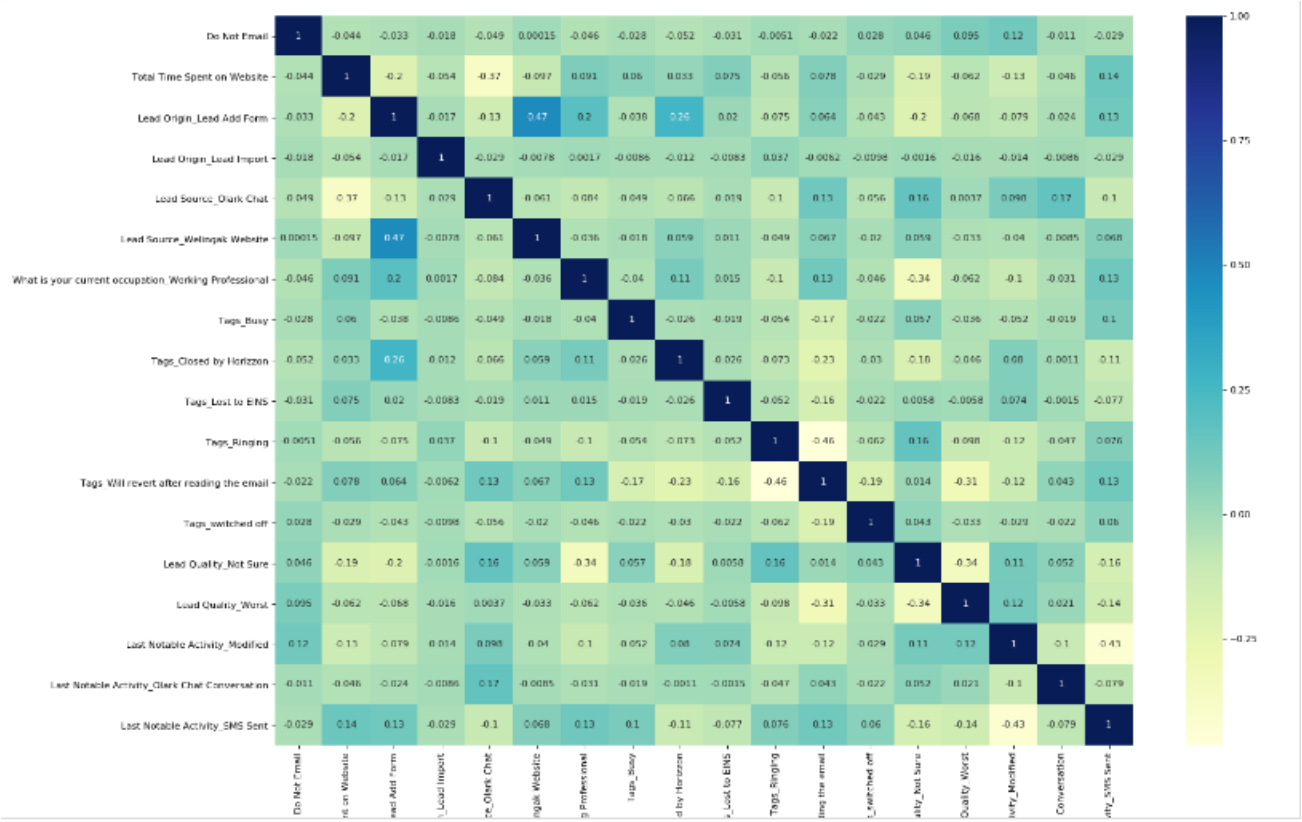
Train-Test data Split:

- We split the initial data into train and test data using the **train_test_split** from **sklearn**
- Splitting the data in accordance to the train_size and test_size which we have considered as **0.7** and **0.3**
- Once the data is split, we perform the **feature scaling** using **StandardScaler** to have a better normalized view of the variables we are dealing with in the dataset.
- We also note that the current conversion rate of the leads is around **38%**

MODEL BUILDING:

- We run the **Generalized Linear model** (GLM) by adding a constant to the train dataset.
- Then we use the **RFE** for **Feature Selection** process to stick with the most significant variables in the dataset.
- We always check for the **VIF values** of the selected variables from RFE.
- The VIF values for all the variables should be minimum and should be always **less than 5**

- Dropping the columns created from **Tags** column and again checking for the VIF values until we reach an optimum value.
- To check if the variables remaining after VIF checks are having **multi-collinearity**, we plot a **heat map** which gives the idea of highly correlated variables



- We find the **probabilities** and the predictions for the train data set and include in a separate column called the **predicted** which gives binary values as **0 or 1**.
- If the **predicted** value is **1** then that customer is having higher probability of Lead conversion

Confusion Matrix:

- A confusion matrix is created to summarize the performance of the algorithm built.
- Here also we created the confusion matrix to summarize the performance of the model we are building and getting the accuracy parameters

Accuracy metrics and beyond for train data set:

- As calculated the accuracy metrics, we have observed that the **overall accuracy** is **92%**
- This means that we have **correctly identified and predicted the significant variables**.
- **Sensitivity** of the model is calculated by the below formula,

  Sensitivity = TP / float (TP+FN) where,

  TP = true positive

  TN = true négatives

  FP = false positives

  FN = false négatives

  Sensitivity of the model is **88%**
- **Specificity** of the model built is calculated as,

  Specificity = TN / float (TN+FP) where,

  TP = true positive

  TN = true négatives

  FP = false positives

FN = false négatives

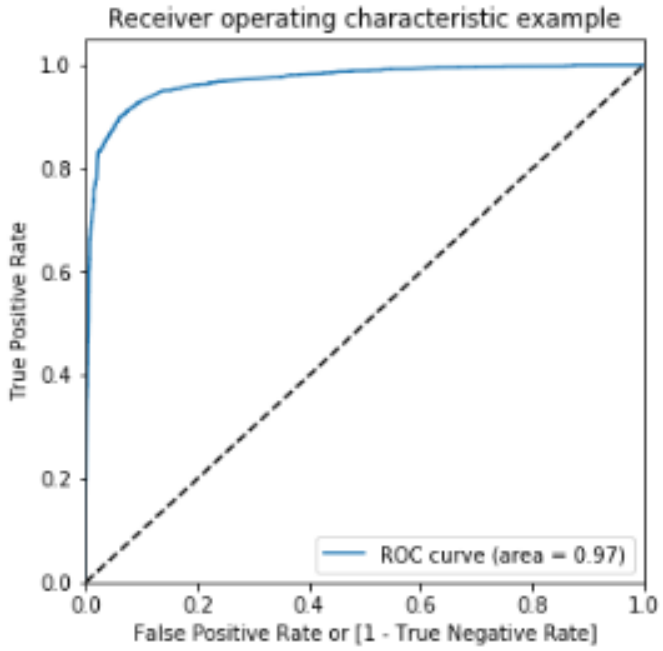
Specificity of the model is **94%**

Plotting the ROC Curve to find the AreaUnderCurve(AUC) :

An ROC curve demonstrates several things:

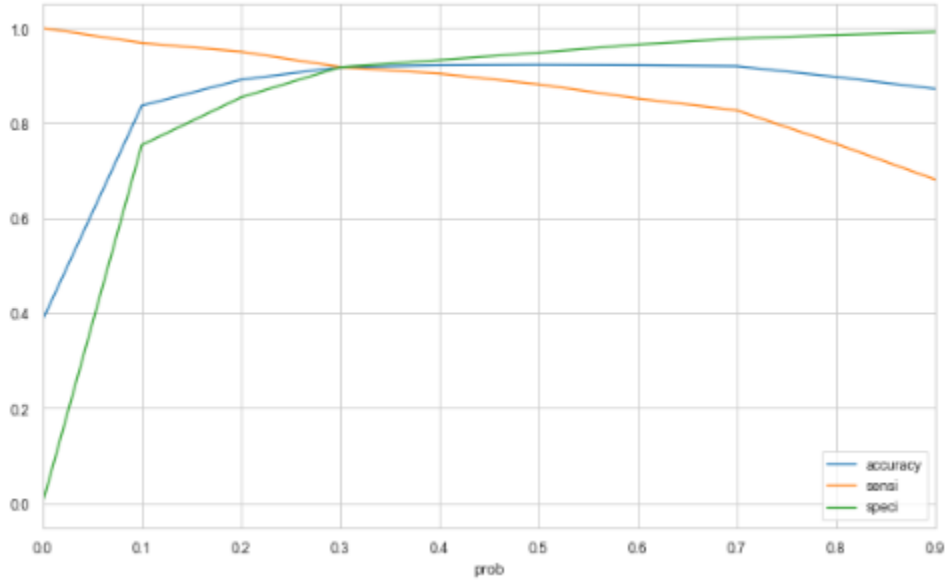
- It shows the **tradeoff between sensitivity and specificity** (any increase in sensitivity will be accompanied by a decrease in specificity).
- The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
- The closer the curve comes to the **45-degree diagonal** of the ROC space, the less accurate the test.

Below is the ROC curve for our model,



From the above we observe that we get a perfect ROC curve which gives the AUC value as **97%**

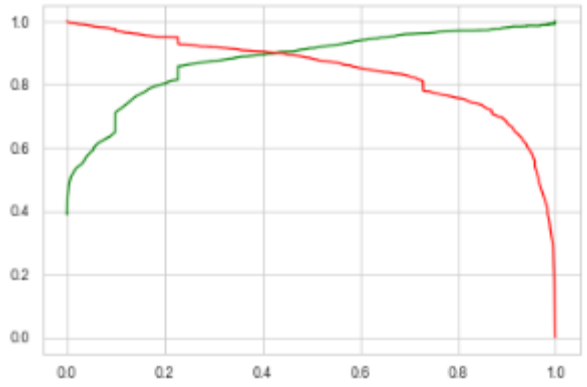
- Now we have to find the optimal cutoff point by taking multiple probabilities points to plot the sensitivity, specificity and accuracy



- From the above curve, we observe that the cutoff probability to be taken for optimum point is **0.3**

Precision and Recall metrics:

- Now we can calculate the precision and Recall values for the train data set which needs to be around the same range as expected between **70-90%**
- When calculated for precision we observed the resultant value is **91%** which is excellent but has to be consistent with the test data
- Recall value corresponds to around **88%** which is also optimum for a dataset to attain the requirement of 80% lead conversion rate.
- We now want to know the optimum threshold using the Precision and Recall thresholds in a graph. When we plot them we got the intersection at **0.42**



- But since we got the expected result in the previous metrics with sensitivity, specificity and accuracy we stick with the same metrics that gave the optimum cutoff as **0.3**

Accuracy metrics and beyond for test data set:

- As calculated the accuracy metrics, we have observed that the **overall accuracy** is **90.7%**
- This means that we have **correctly identified and predicted the significant variables**.
- **Sensitivity** of the model is calculated by the below formula,

  Sensitivity = TP / float (TP+FN) where,

  TP = true positive

  TN = true négatives

  FP = false positives

  FN = false négatives

  Sensitivity of the model is **90.2%**
- **Specificity** of the model built is calculated as,

  Specificity = TN / float (TN+FP) where,

TP = true positive

TN = true négatives

FP = false positives

FN = false négatives

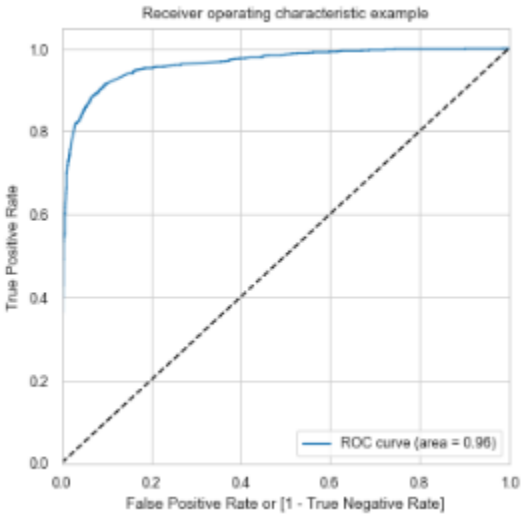
Specificity of the model is **91%**

Plotting the ROC Curve to find the AreaUnderCurve(AUC) :

An ROC curve demonstrates several things:

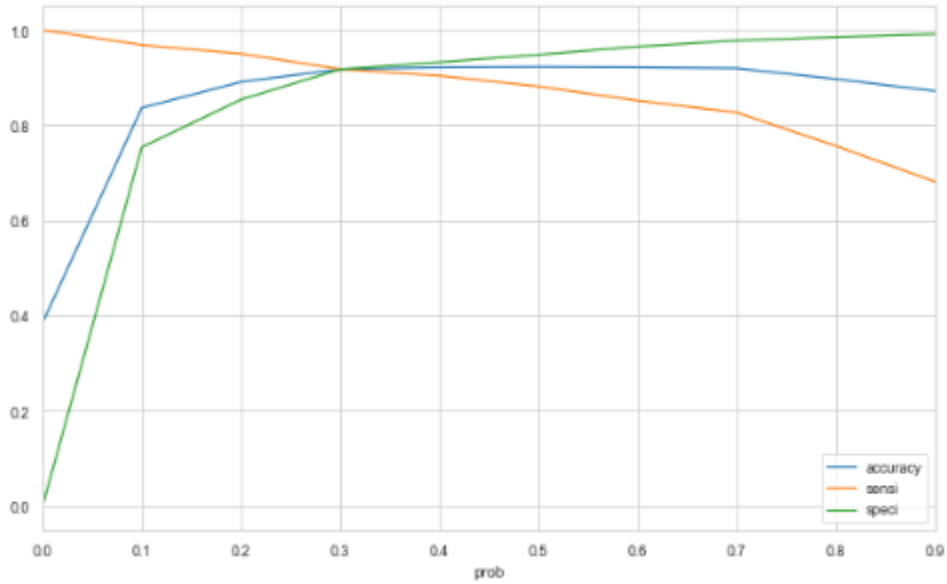
- It shows the **tradeoff between sensitivity and specificity** (any increase in sensitivity will be accompanied by a decrease in specificity).
- The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
- The closer the curve comes to the **45-degree diagonal** of the ROC space, the less accurate the test.

Below is the ROC curve for our test data,



From the above we observe that we get a perfect ROC curve which gives the AUC value as **96.3%**

- Now we have to find the optimal cutoff point by taking multiple probabilities points to plot the sensitivity, specificity and accuracy



- From the above curve, we observe that the cutoff probability to be taken for optimum point is **0.3**

Precision and Recall metrics for test dataset:

- Now we can calculate the precision and Recall values for the test data set which needs to be around the same range as train data set
- When calculated for precision we observed the resultant value is **85.2%** which is excellent but has to be consistent with the test data
- Recall value corresponds to around **90.2%** which is also optimum for a dataset to attain the requirement of 80% lead conversion rate.

Concatenating train and test data to find conversion probabilities:

- After concatenating those leads who have the **final_predict value as 1** and has a lead score more than **39** will be the leads identified as successful Leads in conversion of customers.

Features that are significant to the model evaluation:

COMPARING THE METRICS OF TEST AND TRAIN DATA:

- All metrics are in percentages **(%)**

Dataset	Auc	Sensitivity	Specificity	Accuracy	Precision	Recall
Train	97	88	94	92	91	88
Test	96.3	90.2	91	90.7	85.2	90.2

- All metrics are close to each other both train and test datasets implying that the model is giving excellent results

CONCLUSION AND PREDICTIONS:

The metrics and values such as accuracy, sensitivity, specificity, precision and recall values are very similar and identical for both test and train Datasets implying that the model built is very effective in predicting the Lead conversion and in identifying the Leads with higher conversion rate.

The overall accuracy of **0.9174** at a probability threshold of **0.33** on the test dataset is also very acceptable.

Based on our model, some features are identified which contribute most to a Lead getting converted successfully.

The below mentioned features are those having the **positive** coefficients.

So the conversion probability of a lead increases with increase in

Values of the following features in descending order:

Tags_Lost to EINS

Tags_Closed by Horizzon

Tags_Will revert after reading the email

Tags_Busy

Lead Source_Welingak Website

Lead Origin_Lead Add Form

Last Notable Activity_SMS Sent

Lead Origin_Lead Import

What is your current occupation_Working Profes

Total Time Spent on Website

Lead Source_Olark Chat

The below are the features with **negative** coefficients.

So the conversion probability of a lead increases with decrease in values of the following features in descending order

Last Notable Activity_Modified

Do Not Email

Last Notable Activity_Olark Chat Conversation

Tags_Ringing

Tags_switched off

Lead Quality_Not Sure

Lead Quality_Worst